

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

CLASSIFICATION, REPRÉSENTATION ET SPÉCIALISATION DES
PROCESSUS D’AFFAIRES POUR LE DÉVELOPPEMENT DE SYSTÈMES
D’INFORMATION

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE

PAR
ABDERRAHMANE LESHOB

MAI 2013

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

À la mémoire de mon père

REMERCIEMENTS

Mes sincères remerciements vont tout d'abord à mon directeur Hafedh Mili, professeur à l'Université du Québec à Montréal, pour m'avoir soutenu tout au long de ce projet par sa disponibilité, son expertise, ses idées, ses conseils ainsi que sa patience et son soutien et sans qui cette thèse n'aurait jamais vu le jour.

Je remercie les membres du jury d'avoir accepté d'évaluer cette thèse. J'aimerais également remercier Olivier Gerbe, professeur à l'école des Hautes Études Commercialè de Montréal, Benoit Charbonneau, conseiller principal en processus d'impartition chez AON Hewitt Canada, ainsi que Ait-Idir Mustapha, architecte à la Banque Nationale du Canada, d'avoir participé à la validation de l'approche proposée dans ce travail.

Je tiens également à remercier Anis Boubaker, collègue du laboratoire LATECE, d'avoir consacré du temps pour réviser ma thèse.

Je souhaite aussi exprimer ma reconnaissance à la compagnie AON Hewitt Canada pour sa contribution financière à ce projet de thèse.

Sur un plan plus personnel, je ne peux oublier de remercier ma femme Narimane pour son amour sincère, sa patience, son soutien indéfectible, ses encouragements inconditionnels et sa compréhension. Merci d'avoir pris soin de nos trois enfants, Sarah, Sami et Imad pendant les moments les moins faciles. Aussi, merci à ma petite Sarah, qui se demandait pourquoi son papa est toujours à l'étude, pour ses beaux dessins d'encouragement.

Je remercie aussi toute ma famille. Tout spécialement, je remercie ma mère pour son amour et son soutien, ainsi que mon père, décédé en 2011, qui m'a énormément donné et m'a toujours expliqué que l'avenir passe par les études. Aussi, j'adresse mes plus sincères remerciements à mes frères et mes sœurs pour leurs encouragements.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, qu'ils trouvent ici mes plus vifs remerciements.

TABLE DES MATIÈRES

LISTE DES FIGURES	xiii
LISTE DES TABLEAUX	xix
LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES	xxi
RÉSUMÉ	xxiii
CHAPITRE I	
INTRODUCTION	1
1.1 Problématique	1
1.1.1 Développement par la réutilisation.....	2
1.1.2 Développement par la transformation	3
1.2 La réutilisation au niveau des processus d'affaires	4
1.2.1 Qu'est-ce qu'un processus d'affaires	4
1.2.2 Modélisation de processus d'affaires	5
1.2.3 Classification de processus d'affaires	6
1.2.4 Spécialisation de processus d'affaires	6
1.3 Objectifs de la recherche.....	6
1.4 Organisation de la thèse	7
1.5 Synthèse	8
CHAPITRE II	
ÉTAT DE L'ART	11
2.1 Processus d'affaires.....	11
2.2 Langages de modélisation des processus d'affaires	13
2.2.1 Langages traditionnels.....	14
2.2.2 Langages dynamiques	16
2.2.3 Langages d'intégration de processus.....	21
2.2.4 Langages de modélisation orientée objet	25
2.2.5 Les ontologies d'affaires	25
2.2.6 Comparaison des langages de modélisation des processus d'affaires	30
2.3 Initiatives de réutilisation de processus d'affaires	31
2.3.1 La réutilisation par les patrons d'analyse	31

2.3.2	La réutilisation par les patrons d'affaires	33
2.3.3	La réutilisation par les cadres d'application	35
2.3.4	La réutilisation par l'approche BPM	42
2.3.5	La réutilisation à base de répertoire de processus d'affaires	45
2.4	L'approche de spécialisation par les questions.....	47
2.5	Conclusion.....	48
CHAPITRE III		
VUE GLOBALE DE NOTRE APPROCHE DE SPÉCIFICATION DE PROCESSUS D'AFFAIRES		
		51
3.1	Une nouvelle approche de spécification générique	52
3.1.1	Principes	52
3.1.2	Exemple.....	54
3.2	L'approche métalangage pour la spécification de processus d'affaires	56
3.2.1	Principes	56
3.2.2	Qu'est-ce qu'un processus d'affaires générique.....	58
3.2.3	Représentation de processus d'affaires.....	59
3.2.4	Représentation des questions.....	60
3.2.5	Représentation des spécialisations.....	62
3.2.6	Implémentation.....	63
3.2.7	Validation de l'approche métalangage	67
3.2.8	Le problème de base de l'approche métalangage	70
3.2.9	Vers une approche à base d'une ontologie métier	72
3.3	L'approche ontologique de spécification de processus d'affaires.....	73
3.3.1	Principes	73
3.3.2	Le choix d'une ontologie d'affaires.....	74
3.3.3	Démarche de spécification de processus d'affaires	75
3.3.4	Exemple : Spécification du processus d'approvisionnement	75
3.4	Notre approche de validation.....	88
3.5	Conclusion.....	88
CHAPITRE IV		
L'ONTOLOGIE D'AFFAIRES REA		
		91
4.1	Motivation pour l'approche ontologique.....	91

4.1.1	Le choix de l'ontologie d'affaires REA	93
4.1.2	L'ontologie d'affaires REA.....	95
4.2	Modélisation de processus d'affaires avec REA.....	102
4.2.1	Le patron de processus d'échange REA.....	103
4.2.2	Le patron de processus de conversion REA.....	106
4.2.3	La perspective REA de la modélisation des processus d'affaires	108
4.2.4	La chaîne de valeur REA.....	109
4.3	Extension du méta-modèle de REA	113
4.3.1	Le concept d'engagement.....	113
4.3.2	Le concept de contrat	115
4.3.3	La typification	116
4.3.4	Les extensions des événements REA	118
4.4	Conclusion.....	119
CHAPITRE V		
REPRÉSENTATION ET CLASSIFICATION DES PROCESSUS D'AFFAIRES		121
5.1	Problématique de la représentation de processus génériques.....	121
5.1.1	Qu'est-ce qu'un processus d'affaires générique ?.....	122
5.1.2	Une vue REA ?.....	124
5.1.3	Comment représenter un processus générique ?	125
5.1.4	L'analyse du processus générique.....	126
5.1.5	La conception du processus générique.....	128
5.1.6	Exemple : représentation du processus d'affaires de vente et distribution.....	129
5.2	Implémentation	141
5.2.1	Le cadre de modélisation EMF	142
5.2.2	Le méta-modèle REA de base	144
5.2.3	Le méta-modèle REA étendu	146
5.2.4	Le méta-modèle comportemental	147
5.2.5	Implémentation des vues de processus d'affaires	148
5.2.6	Exemple : la vue REA du processus de vente et distribution.....	149
5.3	Vers un catalogue de processus d'affaires génériques	150
5.3.1	Notre approche de classification de processus génériques.....	150

5.3.2	Phase de découverte du processus le plus proche.....	152
5.3.3	Mise en œuvre du catalogue	152
5.4	Conclusion.....	153
CHAPITRE VI		
TRANSFORMATION DE PROCESSUS D’AFFAIRES.....		155
6.1	Une nouvelle approche de spécialisation par les questions	155
6.1.1	Identification des questions génériques	156
6.1.2	Les patrons REA et les questions génériques.....	156
6.1.3	Représentation des questions.....	165
6.1.4	Implémentation des questions	167
6.2	Les transformations de spécialisation.....	168
6.2.1	Principes	168
6.2.2	Exemple : Spécialisation du processus de vente et distribution	176
6.3	Implémentation des règles de transformation.....	193
6.3.1	JBoss Drools pour la représentation des règles	193
6.3.2	Le flux des transformations	197
6.4	Traitement de transformations en chaîne.....	200
6.4.1	Doit-on générer des modèles intermédiaires?.....	201
6.4.2	Les transformations chevauchantes : Problème ouvert.....	202
6.5	Conclusion.....	202
CHAPITRE VII		
EXPÉRIMENTATION ET VALIDATION		205
7.1	Aspects à valider.....	206
7.2	Données expérimentales	207
7.3	Validation de la représentation de processus d’affaires	208
7.3.1	Vérification de la capacité de représentation.....	209
7.3.2	Validation de la correction des modèles	211
7.4	Validation de l’applicabilité des questions génériques.....	214
7.4.1	Identification des questions génériques	215
7.4.2	Généricité/Pertinence des questions génériques	218
7.5	Validation de l’aspect transformationnel.....	220
7.5.1	La correction syntaxique des transformations	221

7.5.2	La correction sémantique des transformations	222
7.5.3	La confluence des transformations.....	223
7.6	Conclusion.....	227
CONCLUSION ET PERSPECTIVES		229
APPENDICE A		
LES QUESTIONS DE SPÉCIALISATION DE Petia Wohed.....		239
APPENDICE B		
LE MÉTA-MODÈLE EMF		241
APPENDICE C		
NOTRE MÉTA-MODÈLE DE PROCESSUS D’AFFAIRES		243
APPENDICE D		
LA VUE REA DU PROCESSUS DE VENTE ET DISTRIBUTION		247
BIBLIOGRAPHIE		251

LISTE DES FIGURES

Figure	Page
1.1 Un processus d'affaires générique. (Source : Magal et Word, 2010).....	5
2.1 Le méta-modèle de WFMS. (Source : WFMC, 1999).....	17
2.2 Un exemple de messages en BPEL.	19
2.3 La hiérarchie de modélisation d'OMG. (Source : OMG, 2008).....	20
2.4 Structure de la norme RosettaNet. (Source : RosettaNet, 2010)	22
2.5 Processus d'émission de bon de commande. (Source : RosettaNet, 2011)	23
2.6 Scénario de collaboration avec ebXML. (Source : Mili <i>et al.</i> , 2010).....	24
2.7 Le méta-modèle REA de base. (Source : McCarthy, 1982)	27
2.8 Un échange avec l'ontologie e ³ -Value. (Source : Schuster et Motal, 2009).....	28
2.9 Les composants de l'ontologie e-BMO. (Source : Osterwalder et Pigneur, 2002)	29
2.10 Le patron « Participant-Transaction ». (Source : Coad, North et Mayfield, 1997)	32
2.11 Le patron Transaction. (Source : Fowler, 1997).....	32
2.12 Le patron d'application « point-à-point ». (Source : Adams <i>et al.</i> , 2009).....	34
2.13 Le patron « Contrat ». (Source : Eriksson et Penker, 2000).....	35
2.14 Les archétypes et leurs couleurs. (Source : Coad, Lefebvre et De Lucca, 1999)	36
2.15 Les composants du processus de vente. (Source : Coad, Lefebvre et De Lucca, 1999) ..	37
2.16 Le cadre IBM San Fransisco. (Source : Rubin, Christ et Bohrer, 1998)	39
2.17 Exemple de processus de base. (Source : Hallerbach, Bauer et Reichert, 2010).....	40
2.18 Exemple d'options du cadre Provoq. (Source : Hallerbach, Bauer et Reichert, 2010) ...	41
2.19 Le processus obtenu après l'application de l'option 3. (Source : Hallerbach, Bauer et Reichert, 2010)	41
2.20 Les catégories des normes en relation avec le cadre BPM. (Source : Ko, 2009)	43
2.21 L'approche SBPM. (Source : Hepp et Roman, 2007)	44
2.22 Boussole de processus du manuel MIT (Source: Malone, Crowston et Herman, 2003).45	
2.23 Exemple de spécialisation au sein du manuel MIT. (Source : Malone <i>et al.</i> , 1999)	46
3.1 Les étapes de notre approche.....	53
3.2 La vue fonctionnelle simplifiée du processus d'approvisionnement.....	54

3.3	La vue informationnelle du processus d'approvisionnement.	55
3.4	La vue informationnelle après la spécialisation de l'entité produit.	56
3.5	Un méta-modèle primaire de processus d'affaires. (Source : Mili <i>et al.</i> , 2009)	57
3.6	Le modèle des vues d'un processus d'affaires.....	59
3.7	Le modèle de représentation des questions.....	61
3.8	Le modèle de spécialisation des processus d'affaires.....	63
3.9	Une vue simplifiée de notre méta-modèle de processus d'affaires.....	64
3.10	Exemple de représentation d'une question générique.	65
3.11	Exemple de règle de transformation.	66
3.12	La vue informationnelle (modèle Ecore) après la transformation.	67
3.13	Modèle UML simple.....	71
3.14	Les niveaux de classification des concepts.....	72
3.15	Processus d'approvisionnement.....	76
3.16	Une version simplifiée du patron contrat. (Source : Hruby, 2006).....	77
3.17	La chaîne de valeur du processus d'approvisionnement.....	78
3.18	La vue REA étendue du processus d'approvisionnement.....	80
3.19	La vue organisationnelle du processus d'approvisionnement.....	81
3.20	La vue dynamique du processus d'approvisionnement.	81
3.21	La vue informationnelle du processus d'approvisionnement.	82
3.22	La vue REA après la transformation.....	85
3.23	La vue organisationnelle après la transformation.	85
3.24	La vue dynamique après la transformation.....	86
3.25	La vue informationnelle après la transformation.	87
3.26	Le processus d'approvisionnement final après la transformation.....	87
4.1	L'ontologie REA de base. (Source : McCarthy, 1982).....	97
4.2	La situation économique avant l'échange économique. (Source : McCarthy, 2004)	98
4.3	La situation économique après le processus de vente. (Source : McCarthy, 2004).....	98
4.4	Le modèle REA de la vente vu par Elmo. (Source : McCarthy, 2004)	99
4.5	La situation économique après le processus de paiement. (Source : McCarthy, 2004)	100
4.6	Le modèle REA complet de l'échange vu par Elmo. (Source : McCarthy, 2004).....	101
4.7	Le modèle REA du point de vue d'agent indépendant (Source : McCarthy, 2004)	102

4.8 Le méta-modèle REA de base en UML. (Source : Geerts et McCarthy, 2000)	103
4.9 Le patron d'échange REA. (Source : Hruby, 2006)	104
4.10 Un exemple de dualité d'échange.....	105
4.11 Le patron de conversion REA. (Source : Hruby, 2006)	106
4.12 Un exemple de dualité de conversion.....	108
4.13 Le processus d'échange du point de vue des partenaires d'affaires.	109
4.14 Exemple de contexte d'affaires d'une entreprise. (Source : Hruby, 2006).....	110
4.15 La chaîne de valeur initiale. (Source : Hruby, 2006).....	111
4.16 La chaîne de valeur finale (Source : Hruby, 2006).....	112
4.17 L'engagement et l'événement économique en REA. (Source : Hruby, 2006)	114
4.18 Le contrat, l'engagement et le terme. (Source : Hruby, 2006)	115
4.19 L'ontologie étendue de REA. (Source : Hruby, 2006)	116
4.20 Exemple de relation de caractérisation. (Source : Geerts et McCarthy, 2002).....	117
5.1 Le processus générique minimal.	124
5.2 Processus de représentation du processus générique.....	126
5.3. Les vues d'un processus d'affaires.....	129
5.4 Le processus de vente et distribution.....	129
5.5 La chaîne de valeur du processus de vente et distribution.....	131
5.6 Le modèle REA de base du processus de vente.	132
5.7 Le modèle REA de base du processus de distribution.....	133
5.8 Le modèle REA de base du processus de vente et distribution.	134
5.9 La vue REA finale du processus de vente.	135
5.10 La vue REA finale du processus de distribution.	136
5.11 La vue organisationnelle du processus de vente et distribution.	137
5.12 La vue dynamique du processus de vente et distribution.	138
5.13 La vue informationnelle du processus de vente.....	139
5.14 La vue informationnelle du processus de distribution.....	140
5.15 Hiérarchie des classes du packaging Ecore du cadre EMF. (Source : Eclipse, 2008) ...	144
5.16 La mise en œuvre de l'ontologie REA.	145
5.17 Notre méta-modèle REA étendu.....	146
5.18 Notre méta-modèle de la vue dynamique.....	147

5.19 Le modèle des vues d'un processus d'affaires.....	148
5.20 La vue REA du processus de vente et distribution.	149
5.21 La structure de notre référentiel.....	151
5.22 Mise en œuvre de notre catalogue de processus d'affaires.....	153
6.1 Le patron d'affaires « Engagement ». (Source : Hruby, 2006).....	157
6.2 Le patron d'affaires « Contrat ». (Source : Hruby, 2006).....	160
6.3 La chaîne de valeur du processus de conversion initial.....	161
6.4 La nouvelle chaîne de valeur après la sous-traitance.	162
6.5 Le modèle REA de base du processus de conversion initial.....	162
6.6 Le modèle REA final après la sous-traitance.....	163
6.7 Le concept de créance. (Source : Hruby, 2006).....	164
6.8 Le patron d'affaires « Matérialisation de créance ». (Source : Hruby, 2006).....	164
6.9 Notre modèle de représentation des questions.....	166
6.10 Exemple de représentation d'une question générique.	167
6.11 Le processus de spécialisation.	170
6.12 Le modèle de spécialisation de processus d'affaires.	171
6.13 Le processus de vente et distribution.....	177
6.14 La représentation de la question de sous-traitance.	178
6.15 La représentation de la question du calcul automatique des créances.	178
6.16 La nouvelle chaîne de valeur après la sous-traitance.....	179
6.17 La vue REA de base du processus d'achat du service de distribution.....	180
6.18 La vue REA complète du processus d'achat du service de distribution.....	181
6.19 Le processus de distribution après la sous-traitance.....	182
6.20 Le modèle REA après la sous-traitance de la distribution.....	182
6.21 La vue REA finale du processus de la vente.....	183
6.22 La vue REA finale du processus d'achat du service de distribution.....	184
6.23 La vue organisationnelle finale du processus de vente et distribution.....	185
6.24 La vue dynamique finale du processus de vente et distribution.....	186
6.25 Le processus de vente et distribution après la sous-traitance de la distribution.....	187
6.26 La vue informationnelle du processus d'achat du service de distribution.....	189
6.27 La vue informationnelle finale du processus d'achat du service de distribution.....	189

6.28 La vue informationnelle finale du processus de vente.....	190
6.29 La vue informationnelle finale du processus de distribution.....	191
6.30 Le nouveau processus d'affaires de vente et distribution.....	192
6.31 Exemple de règle de transformation de la vue informationnelle.....	195
6.32 Exemple de règle de transformation de la vue REA.....	196
6.33 Exemple EMF : la méthode <code>removeEventsByTypeName</code>	197
6.34 Le flux de la séquence T^{rea+} avec Drools Flow.....	198
6.35 Le flux de la séquence T^{rea-} avec Drools Flow.....	198
6.36 Extrait du code qui permet de charger les règles et le flux T^{rea}	199
6.37 Extrait du code permettant le déclenchement du processus de spécialisation.....	200
7.1 Les processus d'affaires clés des systèmes PGI. (Source : Magal et Word, 2010)	207
7.2 Les événements économiques dans un processus d'échange. (Source : Hruby, 2006)	216
7.3 La vue informationnelle avant l'application des règles $R_{q4, 1}$ et $R_{q5, 0}$	225
7.4 La vue informationnelle après l'application des règles $\langle R_{q4, 1}, R_{q5, 0} \rangle$	226
7.5 La vue informationnelle après l'application des règles $\langle R_{q5, 0}, R_{q4, 1} \rangle$	226
B.1 Le méta-modèle EMF. (Source : Eclipse, 2008).....	242
C.1 Le diagramme EMF de notre méta modèle de processus d'affaires.....	245

LISTE DES TABLEAUX

Tableau	Page
2.1 Comparaison des langages de modélisation de processus (Adapté de Mili <i>et al.</i> , 2010) .	30
7.1 Les processus d'affaires utilisés pour valider notre approche	208
7.2 Résultat de la validation de l'aspect de représentation de processus d'affaires	212
7.3 Résultat de la validation de la correction des modèles obtenus.....	214
7.4 Résultat de la validation de l'applicabilité des questions génériques.....	219

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

BPDM	Business Process Definition Metamodel
BPEL	Business Process Executable Language
BPM	Business Process Management
BPML	Business Process Modeling Language
BPMN	Business Process Model and Notation
e-BMO	e-Business Model Ontology
EbXML	Electronic Business eXtensible Markup Language
EMF	Eclipse Modeling Framework
EPC	Event-Driven Process Chains
MDA	Model Driven Architecture
MOF	Meta Object Facility
OMG	Object Management Group
PGI	Progiciels de Gestion Intégrés
PIM	Platform Independent Model
PSM	Platform Specific Model
RAD	Role Activity Diagram
REA	Resource Event Agent
UML	Unified Modeling Language
UMM	UN/CEFACT Modelling Methodology
XMI	XML Metadata Interchange
XPDL	XML Process Definition Language

RÉSUMÉ

L'amélioration de la productivité et de la qualité des systèmes d'information représente une importante préoccupation en ingénierie logicielle (Fenton et Pfleeger, 1998). Dans ce contexte, la réutilisation a été reconnue comme une solution possible permettant la construction rapide et à moindre coût de logiciels de meilleure qualité (Pressman, 2005). La recherche en réutilisation de logiciel a réalisé beaucoup de progrès. Cela a permis de réutiliser des artéfacts logiciels de différents niveaux d'abstraction et de granularité tels que des bibliothèques logicielles, des patrons de conception, des cadres d'application, voire même des composants d'affaires. Or, nous croyons que la réutilisation peut commencer encore plus tôt, à savoir au niveau des besoins d'affaires. En effet, dans la mesure où les entreprises développent des systèmes d'information pour supporter leurs processus d'affaires, on peut s'attendre à identifier des opportunités de réutilisation en amont, au niveau des processus d'affaires. Mais est-ce vraiment le cas ? Il est vrai que certains processus d'affaires sont spécifiques à une industrie ou à une organisation. Cependant, la plupart sont communs à plusieurs industries et peuvent être utilisés modulo quelques modifications dans différents contextes (Mili *et al.*, 2009). Par exemple, la plupart des processus de la chaîne de valeur (Porter, 1985) dépendent seulement du modèle d'affaires des organisations. Ces processus peuvent donc s'appliquer dans différentes industries.

Dans cette thèse, nous proposons une nouvelle méthodologie et des outils pour la réutilisation et l'adaptation de processus d'affaires pour les besoins spécifiques des organisations. Nous visons une approche facile à assimiler par des analystes d'affaires. Notre approche permet la spécialisation de processus d'affaires par le passage d'un modèle générique à un modèle spécifique à l'organisation. Le processus générique est choisi à partir d'un catalogue de processus d'affaires. Ce dernier ne requiert pas d'être exhaustif car nous proposons une spécialisation dynamique.

Notre méthodologie de spécification de processus d'affaires utilise une approche transformationnelle par des questions. Elle est générique, car elle se base sur des questions génériques et des transformations qui s'appliquent à des processus d'affaires de différents domaines. Nos questions, comme nos transformations sont basées sur des patrons d'affaires qui, à leur tour, sont basés sur l'ontologie d'affaires REA (*Resource Event Agent*) (Geerts et McCarthy, 2000).

Pour valider notre approche, nous l'avons appliquée à différents processus d'affaires provenant de différents domaines. Nous avons effectué des tests pratiques pour vérifier la représentation de processus d'affaires, l'applicabilité des questions et les transformations de spécialisation. Les résultats obtenus montrent que notre approche est prometteuse. Nos travaux mettent également en lumière quelques défis à relever.

Mots-clés : Processus d'affaires, Patron d'affaires, Ontologie d'affaires, Réutilisation, Spécialisation, Chaîne de valeur, Transformation de modèles.

CHAPITRE I

INTRODUCTION

1.1 Problématique

Aujourd'hui, les systèmes d'information sont plus complexes et plus coûteux que ceux des décennies précédentes. Bien que des progrès importants aient été réalisés dans le domaine de l'ingénierie logicielle, ces systèmes continuent à faire face à divers problèmes tels qu'une qualité insatisfaisante, des délais de livraison non respectés ou encore des coûts plus importants que prévu (Wiegers, 2003). En effet, le développement des systèmes d'information demeure une tâche complexe malgré l'évolution continue dans le domaine du développement logiciel et des infrastructures de déploiement. Nous notons, cependant, les progrès remarquables qui ont été enregistrés dans toutes les phases de développement d'un logiciel, notamment dans l'analyse, la conception et la mise en œuvre. Dans ce cadre, nous soulignons l'évolution importante des langages et des paradigmes de programmation, les bonnes pratiques de développement, notamment l'application de patrons à différents niveaux et le recours aux cadres d'application (*frameworks*).

Malgré les progrès réalisés dans le domaine de l'ingénierie logicielle, le défi de réaliser des systèmes de qualité demeure de taille, d'autant plus qu'aujourd'hui, ces systèmes sont de plus en plus complexes (Presmann, 2005). En effet, le passage de la spécification des exigences d'un logiciel à la réalisation demeure une tâche complexe. Pour gérer cette complexité, l'ingénierie logicielle propose deux approches : (i) le développement par la réutilisation et (ii) le développement par la transformation. Les approches pour la réutilisation

se sont concentrées sur la réutilisation des différents artefacts produits lors du développement logiciel tandis que les approches transformationnelles sont plutôt des approches hybrides, i.e. elles visent à réutiliser aussi bien les artefacts générés lors du développement que le processus de développement lui-même (El boussaidi, 2009).

1.1.1 Développement par la réutilisation

La réalisation d'un système d'information de qualité à moindre coût, tout en minimisant le temps de mise en marché, représente un objectif pour toute organisation qui produit de tels systèmes. La réutilisation, en tant qu'attribut de qualité important en ingénierie logicielle, représente une voie intéressante pour atteindre cet objectif. Elle consiste à réutiliser des artefacts existants et modifiables en fonction de nouveaux besoins que l'on veut traiter. La réutilisation logicielle permet de minimiser les coûts de développement tout en augmentant la qualité des systèmes conçus (Presmann, 2005). Elle présente des avantages d'affaires substantiels; la qualité du produit, la productivité et le coût total sont tous améliorés (Presmann, 2005). Le concept de réutilisation peut être appliqué à plusieurs niveaux logiciels. Mili *et al.* (2001) discutent de hiérarchies de niveaux de réutilisation :

1. *Réutilisation au niveau du code* : c'est le plus bas niveau de réutilisation qui consiste à réutiliser une partie du code, souvent des bibliothèques de fonctions ou de classes. Ce type de réutilisation présente un avantage minimal (Mili *et al.*, 2001).
2. *Réutilisation au niveau de la conception* : elle consiste principalement à réutiliser des modèles de conception comme les patrons de conception et les cadres d'application. L'utilisation des patrons de conception est devenue très populaire depuis la publication des différents patrons par le groupe Gamma *et al.* (1995).
3. *Réutilisation au niveau d'analyse* : à un plus haut niveau, ce type de réutilisation est basé sur les patrons d'analyse et les modèles génériques d'analyse obtenus par l'analyse du domaine. L'utilisation des patrons d'analyse offre deux grands avantages (Geyer-Shulz et Hahsler, 2001). D'abord, elle accélère le développement des modèles d'analyse. Ensuite, elle facilite la transformation de ces modèles vers des modèles de conception. Parmi les travaux qui ont traité ce sujet, nous citons les patrons d'analyse de Fowler (1997) et de Coad, North et Mayfield (1997).

1.1.2 Développement par la transformation

Au lieu de réutiliser des artefacts de développement, l'approche transformationnelle met l'accent sur le processus de développement lui-même. Dans cette approche, le processus de développement est considéré comme une séquence de transformations appliquées à des exigences utilisateur pour aboutir à une mise en œuvre en code exécutable spécifique à une plateforme et conforme aux exigences de qualité. L'approche transformationnelle basée sur les modèles consiste en un raffinement continu de modèles passants de modèles abstraits vers des modèles plus concrets. Parmi les exemples d'approches transformationnelles, nous citons les travaux de Shlaer et Mellor (1992) qui proposent une méthode transformationnelle pour la conception de systèmes orientés objet et les travaux de Lano et Bicarregui (1999) qui proposent des techniques de transformation de diagrammes de classes et d'états-transitions.

L'approche transformationnelle a été reprise par l'OMG (*Object Management Group*) avec l'architecture MDA (*Model Driven Architecture*) (OMG, 2003a). MDA se veut être indépendante de toute plate-forme. L'architecture MDA se base sur le fait qu'un système soit représenté par plusieurs modèles à différentes étapes de son cycle de construction. Ces modèles peuvent être obtenus par transformation ou fusion d'autres modèles. MDA définit une architecture structurée en trois types de modèles : (i) les modèles indépendants des traitements (*CIM, Computation Independent Model*), (ii) les modèles indépendants des plates-formes (*PIM, Platform Independent Model*), et (iii) les modèles spécifiques aux plates-formes (*PSM, Platform Specific Model*) (OMG, 2003a).

Le modèle CIM présente les exigences d'affaires du domaine. Le modèle PIM décrit les fonctions d'un système. Il se concentre sur la modélisation de la logique d'affaires indépendamment de toute plate-forme. Le modèle PSM combine les spécifications du PIM avec des détails spécifiques à une plate-forme de déploiement.

Notre travail de recherche s'inscrit dans le cadre d'un programme de recherche qui vise le développement de systèmes d'information par transformations de modèles du domaine d'affaires et, plus particulièrement, des modèles des *processus d'affaires* que le système d'information doit supporter. Or, nous croyons que pour pouvoir générer des modèles logiciels à partir de modèles de processus d'affaires, il nous faut des modèles de processus

assez détaillés, qui vont au-delà de la capacité technique et de la patience de l'analyste d'affaires typique. Il nous faut donc offrir aux analystes des outils qui leur permettent de générer des processus spécifiques à leur organisation à partir d'un catalogue de processus génériques et d'une connaissance d'assez haut niveau de certains aspects clés de leur processus d'affaires/métier.

1.2 La réutilisation au niveau des processus d'affaires

La plupart des processus d'affaires dépendent du modèle d'affaires et non du domaine d'affaires spécifique des entreprises. En pratique, les modèles de processus d'affaires sont communs à plusieurs industries. Dès lors, les modèles de ces processus peuvent être réutilisés à travers plusieurs industries (Mili *et al*, 2009). Dans ce contexte, il serait judicieux de pouvoir réutiliser des processus d'affaires ainsi que les services logiciels associés. De là, l'objectif de notre recherche vise une approche permettant de réutiliser des modèles de processus d'affaires génériques et de les transformer pour les adapter aux besoins spécifiques des organisations.

1.2.1 Qu'est-ce qu'un processus d'affaires

Un processus d'affaires est un ensemble d'étapes réalisées par un ou plusieurs agents pour accomplir un objectif spécifique pour lequel le processus a été conçu (Curtis, Kellner et Over, 1992). Un processus d'affaires *générique* est un processus d'affaires qui est indépendant de tout contexte d'industrie spécifique. La Figure 1.1 montre un processus d'affaires générique d'après la vision des progiciels de gestion intégrés (PGI) (Magal et Word, 2010). Chaque processus est déclenché par un ou plusieurs événements. Les différentes étapes du processus sont complétées dans différents domaines fonctionnels. Par exemple, pour un processus de vente, c'est le département des ventes qui reçoit et traite l'ordre d'achat du client, le département de distribution expédie les produits et, finalement, le département de comptabilité traite les factures et les paiements.

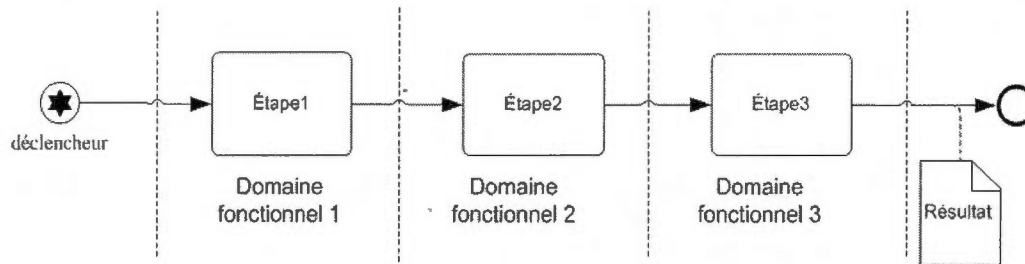


Figure 1.1 Un processus d'affaires générique. (Source : Magal et Word, 2010)

Pour réaliser notre objectif de réutilisation et d'adaptation de processus d'affaires génériques, nous devons : (i) définir une représentation (i.e. modélisation) explicite du processus d'affaires pour pouvoir le manipuler, (ii) proposer une organisation (i.e. classification) des processus dans un catalogue permettant de faciliter la recherche et la navigation, et (iii) développer un ensemble de techniques permettant de spécialiser un processus générique pour l'adapter aux besoins spécifiques des organisations.

1.2.2 Modélisation de processus d'affaires

Un modèle est une vue simplifiée d'une réalité complexe. Cette vue consiste à créer des abstractions qui permettent d'éliminer les détails non pertinents et de mettre l'accent sur les aspects les plus importants (Eriksson et Penker, 2000). Le terme « Modélisation de processus d'affaires » a été inventé dans les années 1960 dans le domaine de l'ingénierie des systèmes. La modélisation de processus d'affaires permet de les décrire, de les analyser et de les mettre en œuvre (Ould, 1995). La modélisation de processus d'affaires a suscité l'intérêt de plusieurs chercheurs et comités de normalisation. Les chercheurs et les praticiens dans le domaine ont longuement reconnu que la compréhension du processus d'affaires qu'un système doit supporter est la clé de l'identification des besoins des clients (Mili *et al.*, 2010).

Phalp et Shepperd (2000) considèrent la phase d'identification et de compréhension des besoins clients, lors du développement des systèmes d'information, comme une étape très importante en ingénierie logicielle qui demeure pourtant très problématique. Des études menées par Davis (1993) montrent que de 40 à 60% des défauts découverts dans les systèmes sont des erreurs liées à la compréhension des exigences. Phalp et Shepperd (2000)

préconisent la modélisation préalable des processus d'affaires comme une solution pour comprendre les besoins des entreprises et le contexte d'utilisation des systèmes. En effet, pour comprendre ce qu'un logiciel est censé faire, nous devons le mettre dans le cadre du processus d'affaires qu'il doit supporter.

1.2.3 Classification de processus d'affaires

La classification décrit une propriété d'un concept ou sa relation avec d'autres concepts afin d'établir le regroupement sous forme de classes (UN/CEFACT, 2005). En général, elle permet une organisation de différentes entités sous forme de classes reliées. La classification de processus d'affaires permet, notamment, de faciliter l'identification et la recherche d'un processus d'affaires parmi un ensemble de processus. Notre objectif est de fournir une classification qui offre un support simple et structuré permettant de sélectionner les modèles de processus les plus proches de ce que nous voulons réutiliser.

1.2.4 Spécialisation de processus d'affaires

La spécialisation d'un processus d'affaires consiste à produire un nouveau processus d'affaires spécifique à partir d'un autre plus générique. Le processus d'affaires obtenu après la spécialisation doit répondre aux besoins d'affaires spécifiques d'une organisation.

Dans le contexte de notre approche de spécification de processus d'affaires, la spécialisation consiste en l'application d'une série de transformations à un processus générique. Ces transformations, souvent complexes, se basent sur les différents modèles qui représentent les perspectives (ou les vues) du processus d'affaires générique.

1.3 Objectifs de la recherche

La contribution principale de notre thèse est de proposer une nouvelle méthodologie et des outils pour la réutilisation et l'adaptation d'un catalogue de processus génériques. Nous visons une approche de réutilisation facile à assimiler par des analystes d'affaires. Celle-ci préconise le passage d'un modèle générique de processus d'affaires à un modèle spécifique à l'organisation. Le processus générique est choisi à partir d'un catalogue de processus

d'affaires. Le catalogue est organisé selon une hiérarchie qui facilite la navigation et la recherche. Ce dernier ne requiert pas d'être exhaustif. Notre approche de spécialisation automatique de processus d'affaires est générique, car elle se base sur des questions et des transformations génériques.

Cette nouvelle méthodologie s'inscrit dans le cadre des approches de réutilisation hybrides. Elle permet de réutiliser et transformer des modèles de processus d'affaires. Ces modèles sont de type CIM dans la méthodologie MDA (OMG, 2003a). Ils sont fondés sur les vues des processus d'affaires proposées par Curtis, Kellner et Over (1992).

1.4 Organisation de la thèse

Dans le chapitre 2, nous présentons une revue de la littérature qui fait le tour des aspects reliés à notre travail. D'abord, nous donnons une vue globale de différents langages de modélisation des processus d'affaires. Par la suite, nous exposons différentes initiatives de classification et de spécialisation de processus d'affaires. Principalement, nous parlerons : (i) des initiatives à base de patrons d'analyse, (ii) des initiatives à base de patrons d'affaires, et (iii) des initiatives à base de cadres d'application.

Au chapitre 3, nous présentons une vue globale de notre approche de réutilisation de processus d'affaires. D'abord, nous décrivons une première démarche que nous avons adoptée et les raisons ayant mené à son abandon. Par la suite, nous montrons notre nouvelle approche basée sur une ontologie d'affaires, en l'occurrence REA (*Resource Event Agent*). En particulier, nous donnons un aperçu sur notre stratégie de conception et de transformation de processus d'affaires génériques.

Dans le chapitre 4, nous introduisons les concepts les plus importants de l'ontologie d'affaires REA. Nous y détaillons, notamment, les concepts de création et d'échange de valeurs économiques. Nous explicitons également les principaux patrons de processus d'affaires ainsi que les extensions ontologiques de REA qui ont été proposées.

Dans le chapitre 5, nous présentons notre approche de représentation et de classification de processus d'affaires. En premier lieu, nous présentons notre méthodologie de

conception de processus d'affaires génériques au sein d'un catalogue commun. Par la suite, nous décrivons la structure de notre catalogue de processus d'affaires.

Dans le chapitre 6, nous présentons notre approche transformationnelle de processus d'affaires basée sur des patrons d'affaires génériques. Celle-ci permet de réutiliser et de transformer les modèles de processus jusqu'à l'obtention d'un processus qui répond aux besoins d'une organisation. Dans la suite de ce document, nous parlerons d'approche transformationnelle ou tout simplement de réutilisation pour designer notre approche de spécialisation automatique qui vise à réutiliser et transformer les modèles générés lors du processus de spécialisation.

Le chapitre 7 sera dédié à l'expérimentation et à la validation de notre approche. Nous présentons la démarche que nous avons adoptée pour valider les différents aspects de notre méthodologie. Spécifiquement, nous parlerons de la validation de notre représentation de processus d'affaires, de l'application des questions génériques identifiées et de la validation des transformations élucidées. Pour certaines validations, nous avons consulté des experts dans le domaine de processus d'affaires.

Finalement, dans le chapitre 8, nous présentons une synthèse des travaux réalisés et nous évoquons des perspectives futures pour ce travail de recherche.

1.5 Synthèse

Nous avons vu dans ce chapitre que :

1. La réutilisation est un attribut de qualité important pour tout système d'information. Elle peut être appliquée au niveau du code, de la conception et de l'analyse.
2. Les approches transformationnelles pour le développement visent à réutiliser aussi bien les artefacts générés lors du développement que le processus de développement lui-même.
3. La modélisation de processus d'affaires permet de les décrire, de les analyser et de les implémenter (Ould, 1995).
4. La classification de processus d'affaires facilite la recherche d'un processus d'affaires parmi un ensemble de processus.

5. La spécialisation de processus d'affaires permet d'adapter des processus d'affaires génériques aux besoins spécifiques des organisations.

Dans le cadre de ce travail, notre objectif est de développer une nouvelle méthodologie et des outils pour aider l'analyste d'affaires à spécifier de façon précise le processus d'affaires qu'il veut supporter par un système d'information à partir d'un catalogue de processus d'affaires génériques. Ainsi, un analyste d'affaires pourra consulter le catalogue pour trouver le processus qui correspond le mieux aux besoins spécifiques de l'organisation. Si le processus trouvé correspond à ces besoins, il peut le réutiliser tel quel, sinon, il doit l'adapter pour en obtenir un autre. Pour ce faire, nous avons besoin de développer une approche de modélisation, de classification et de spécialisation. (i) La modélisation permet de décrire le processus d'affaires, (ii) la classification facilite l'identification et la recherche d'un processus d'affaires dans un catalogue de processus, et (iii) la spécialisation permet d'obtenir un nouveau processus d'affaires spécifique à partir d'un processus d'affaires plus générique.

Dans le prochain chapitre, nous ferons un état de l'art des différentes méthodes de modélisation, de classification et de spécialisation de processus d'affaires.

CHAPITRE II

ÉTAT DE L'ART

Durant les dernières années, les organisations n'ont cessé de porter un intérêt croissant à l'amélioration de la qualité de leurs processus d'affaires internes et à l'optimisation de leurs échanges avec leurs clients et partenaires d'affaires. (Hallerbach, Bauer et Reichert, 2010). Nous avons montré, dans le chapitre précédent, l'importance des processus d'affaires quant au développement des systèmes d'information. Dans le cadre de cette thèse de doctorat, nous proposons une méthodologie et des outils permettant de réutiliser des modèles de processus d'affaires. Pour ce faire, nous avons besoin de développer une approche de modélisation, de classification et de spécialisation : (i) la modélisation permet de décrire le processus d'affaires, (ii) la classification facilite la recherche d'un processus d'affaires dans un catalogue de processus, et (iii) la spécialisation permet d'adapter un processus générique pour répondre aux besoins spécifiques des organisations.

Dans ce chapitre, nous commençons par présenter une revue de la littérature portant sur les différentes méthodes de modélisation des processus d'affaires. Par la suite, nous montrons les initiatives de spécialisation pour la réutilisation des processus d'affaires.

2.1 Processus d'affaires

Un processus d'affaires est une série d'activités qui prennent une ou plusieurs entrées et qui produisent un résultat ayant une valeur pour le client (Hammer et Champy, 1993). Selon Devenport (1993), un processus d'affaires est un ensemble d'activités structurées et mesurables conçues pour produire un résultat de valeur pour un client ou un marché

particulier. Le processus d'affaires met l'accent sur la façon dont le travail est fait au sein d'une organisation. Les activités sont spécifiques et ordonnées dans le temps et l'espace. Les processus d'affaires ont deux caractéristiques importantes : (i) ils impliquent des personnes internes ou externes à l'organisation, et (ii) ils traversent les frontières de l'entreprise (Devenport, 1993). Dans une autre définition, un processus d'affaires est un ensemble d'activités liées qui prennent une entrée et la transforment pour créer une sortie (Johansson *et al.*, 1993). Idéalement, la transformation doit ajouter une valeur à l'entrée et créer une sortie plus utile pour le destinataire.

Un processus d'affaire peut être défini sur la base de trois dimensions (Davenport et Short, 1990) :

1. *Les entités organisationnelles* : elles peuvent être inter-organisationnelles, inter-fonctionnelles ou interpersonnelles.
2. *Les objets* : le processus manipule des objets. Un objet peut être physique ou informationnel.
3. *Les activités* : c'est l'ensemble des tâches du processus d'affaires. Davenport et Short (1990) distinguent les activités de gestion des activités opérationnelles.

Les processus d'affaires ont deux vues fondamentales et complémentaires : l'orchestration et la chorégraphie (OMG, 2008). L'orchestration décrit la séquence des activités, leurs branchements et leurs synchronisations. La chorégraphie permet de décrire les interactions (i.e. collaborations) entre les partenaires d'affaires.

Il ressort des différentes définitions de processus d'affaires dans la littérature, qu'un processus d'affaires est un ensemble d'*activités* ordonnées et reliées. Ces activités consomment et produisent des *ressources*. Elles sont accomplies par des *acteurs* qui remplissent différents *rôles* dans le contexte d'une *organisation* effectuant des *fonctions* d'affaires. Les activités du processus peuvent être déclenchées par des événements et être reliées par des dépendances de ressources ou des dépendances de contrôle (Mili *et al.*, 2009).

2.2 Langages de modélisation des processus d'affaires

Il existe une grande variété de langages de modélisation de processus d'affaires. En 2005, un doctorant qui a commencé de compiler une liste de langages de modélisation de processus s'est arrêté au 3000^{ème} (Recker, 2005). La majorité des techniques de modélisation ont été développées pour répondre à des besoins spécifiques et sont issues de différentes traditions. Donc, le choix de langage a toujours été dirigé par le besoin. Cette prolifération de langages est une source de confusion (Ko, Lee et Lee, 2009). Selon Curtis, Kellner et Over (1992), la modélisation de processus d'affaires doit faire ressortir les quatre différentes vues (perspectives) suivantes:

- *La vue fonctionnelle* : représente la dépendance fonctionnelle entre les activités du processus. Elle met l'accent sur les activités à accomplir et les ressources produites et consommées par ces activités.
- *La vue dynamique (comportementale)* : montre la séquence des étapes du processus. Les étapes sont des activités ou des éléments de contrôle. Ces derniers décrivent comment les activités sont connectées. La vue dynamique s'intéresse principalement à **quand** et **comment** ces étapes sont connectées.
- *La vue informationnelle*: représente la description structurelle des entités manipulées par les activités du processus d'affaires.
- *La vue organisationnelle*: explicite la structure organisationnelle, les rôles et les mécanismes de communication au sein des entreprises.

Mili *et al.* (2010) proposent une classification des langages de modélisation de processus d'affaires. Ils exposent quatre grandes familles : les langages traditionnels, les langages dynamiques, les langages d'intégration de processus et les langages orientés objet. Nous baserons notre présentation des langages de modélisation de processus d'affaires selon cette classification. Néanmoins, nous y ajouterons les ontologies d'affaires.

2.2.1 Langages traditionnels

Ces langages sont nés à partir des différents courants de modélisation en ingénierie de l'information et des processus.

2.2.1.1 IDEF

IDEF (*Integration DEFinition*) est une famille de langages de modélisation dans le domaine d'ingénierie logicielle. Elle fut créée dans le cadre du projet ICAM (*Integrated Computer Aided Manufacturing*) par l'U.S. Air Force en 1970. Relativement à la modélisation des processus d'affaires, nous distinguons les variantes suivantes (Mili *et al.*, 2010) :

IDEF0 : Créé principalement pour la modélisation fonctionnelle, IDEF0 est un langage fondé sur la notation SADT (*Structured Analysis & Design Technique*). Un modèle IDEF0 consiste en une hiérarchie de diagrammes dont chacun décrit la fonctionnalité d'un système, ses entrées, ses sorties et ses services. Un diagramme est un graphe de nœuds. Un nœud peut être une fonction simple ou un autre diagramme. Les arcs orientés représentent des flux de données ou de contrôle. Une fonction ne peut débuter que lorsque toutes ses entrées sont disponibles.

IDEF1 : Modélise la vue informationnelle qui expose les entités du système ainsi que les relations entre elles. Le langage IDEF1 est basé principalement sur le modèle relationnel de données avec l'approche entité/relation. IDEF1 utilise la technique de conception ENALIM (*Evolving Natural Language Information Model*) connue maintenant sous le nom ORM (*Object Role Modeling*).

IDEF3 : Modélise la vue dynamique en décrivant la séquence des étapes du processus ainsi que les contraintes qui les sous-tendent et les événements du système avec une prise en compte de l'aspect temporel. IDEF3 utilise la terminologie de scénario qui décrit une occurrence d'un processus d'affaires. Un scénario est décrit par deux modèles selon deux stratégies de modélisation. Une première stratégie centrée sur le *processus*; elle décrit les séquences. Une deuxième stratégie centrée sur l'*objet*; elle expose la transition des objets au sein du scénario.

Pour conclure, on remarque qu'IDEF est une norme de modélisation qui permet de spécifier les aspects importants du processus d'affaires. Cependant, nous pouvons remarquer que la vue organisationnelle prévue dans IDEF12 n'est pas encore développée à ce jour.

2.2.1.2 RAD

RAD (*Role Activity Diagram*) est une méthode visuelle pour modéliser et analyser les processus d'affaires (Ould, 1995). Dans le cadre de RAD, un processus d'affaires est un ensemble de rôles exécutés par des acteurs. L'interaction entre ces rôles définit la chorégraphie. Chaque rôle expose l'ensemble des activités ordonnées par leurs états. Une ressource (entité) peut être reliée à une activité ou à une interaction lors d'un échange entre les rôles. RAD expose l'entité sous ses différents états pendant le déroulement du processus en utilisant le concept d'ELH (*Entity Lifetime Histories*). Un diagramme RAD supporte les concepts de base de modélisation de processus d'affaires. Il met l'emphasis sur la vue dynamique et la structure des rôles au niveau organisationnel (Ould, 1995).

2.2.1.3 EPC

EPC (*Event-Driven Process Chains*) est un langage développé à l'institut des systèmes d'information à l'université de Sarland en Allemagne en 1992. EPC offre une notation graphique à base de connecteurs logiques. Il est largement utilisé par la suite logicielle ARIS (*Architecture of Integrated Information Systems*) d'IDS (*Integrated Data-processing Systems*) Scheer et les composants de flux de travaux (*Workflow*) du progiciel SAP R/3 (Ko, Lee et Lee, 2009).

EPC est facile à comprendre par un analyste d'affaires. Ses concepts de base sont les fonctions, les événements et les connecteurs logiques. Un processus EPC est représenté par un graphe d'événements et de fonctions. L'événement décrit la situation avant ou après l'exécution d'une fonction (i.e. une étape du processus). EPC permet de modéliser les processus parallèles. Il est principalement centré sur l'aspect comportemental et fonctionnel d'un processus d'affaires. Dans une version étendue, EPC permet de représenter la vue informationnelle et la vue organisationnelle.

2.2.2 Langages dynamiques

Ces langages partagent les caractéristiques suivantes : (i) ils mettent l'accent sur la vue dynamique selon la classification de Curtis, Kellner et Over (1992), (ii) ils offrent une description complète permettant de mettre en œuvre et exécuter le processus d'affaires, (iii) ils mettent l'accent sur un format de sérialisation pour les échanges, et (iv) ils sont normalisés (Mili *et al.*, 2010).

2.2.2.1 BPMN

Le langage graphique BPMN (*Business Process Model and Notation*) a été présenté en mai 2004 par le consortium BPMI (*Business Process Management Initiative*) qui a fusionné avec l'OMG en 2005. BPMN est un langage graphique récent de plus en plus accepté comme une norme. Avec BPMN, OMG vise à combler le vide entre la technologie de l'information et les analystes d'affaires en présentant une notation graphique simple et facile à assimiler par des utilisateurs ayant moins de connaissance techniques. Avec une base fondée sur les réseaux de Pétri et centrée sur l'aspect comportemental d'un processus d'affaires, BPMN supporte tous les concepts de base d'un processus d'affaires avec une sémantique de flux de contrôle bien définie (Ko, Lee et Lee, 2009).

2.2.2.2 BPML

Comme BPMN, BPMI a présenté BPML (*Business Process Modeling Language*) en 2004 comme le langage qui décrit la représentation structurelle de processus ainsi que la sémantique de son exécution. BPML est une norme pour les systèmes de gestion de processus d'affaires (*Business Process Management System*). BPML offre un langage formel à base d'XML pour représenter des processus exécutables qui traitent tous les aspects des processus d'affaires des entreprises, y compris les activités complexes, les transactions et leur compensation, la gestion des exceptions et la sémantique opérationnelle (BPML, 2003).

Dans BPML, un processus est un ensemble d'activités qui s'exécutent dans un contexte caractérisé par des propriétés spécifiques telles que les variables et les exceptions.

Un processus peut être initié par une activité ou encore par un appel explicite ou par un événement sous forme de signal.

2.2.2.3 XPDL

Le consortium international de gestion de *workflow* (WfMC, *Workflow Management Coalition*) a défini un certain nombre de normes dans le domaine de la gestion des flux. Parmi ces normes, notons le modèle de référence des systèmes de gestion de flux (WfMS, *Workflow Management System*) qui présente le méta-modèle de base pour représenter les systèmes de *workflow* (WfMC, 1999). Ce méta-modèle (fig. 2.1) souligne les éléments de base et les liens qui doivent exister pour supporter l'automatisation d'un processus. Ce méta-modèle vise à établir un format d'échange entre les différents modèles et outils de *workflow*. WfMC propose cinq interfaces qui doivent être supportées par un système de gestion de flux. La première interface propose un langage pour représenter les processus d'affaires.

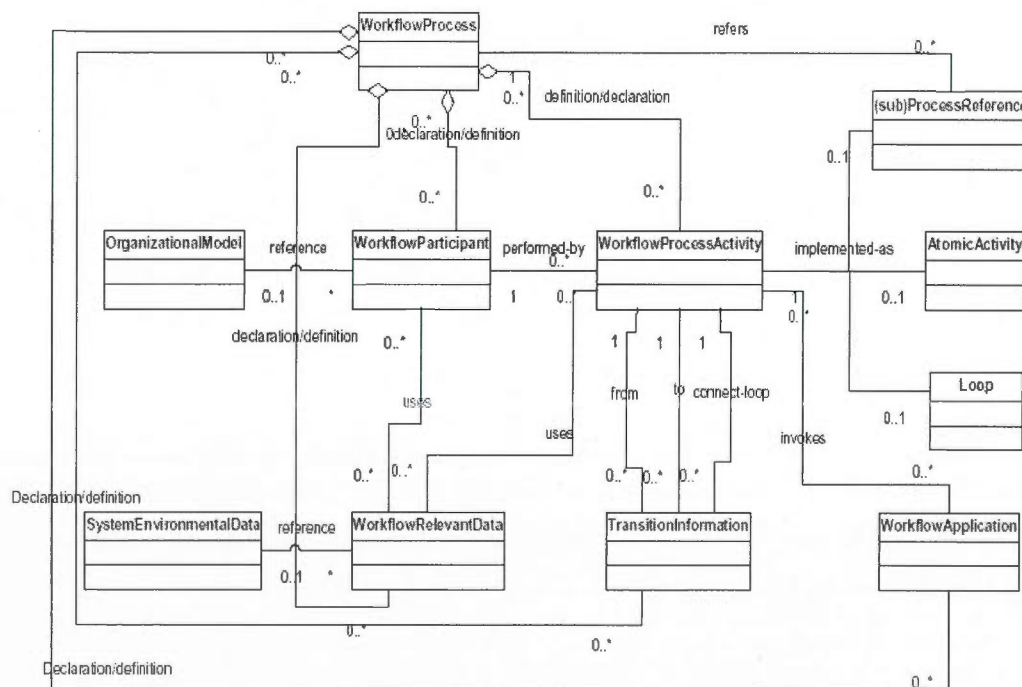


Figure 2.1 Le méta-modèle UML de WfMS. (Source : WfMC, 1999)

Dans une première version, WfMC a proposé le langage WPDL (*Workflow Process Description Language*) (WfMC, 1999). Avec l'émergence du langage XML comme une norme d'échange de données, le WfMC a produit, en 2002, un nouveau langage de définition de processus dit XPDL (*XML Process Definition Language*) basé sur XML (Ko, Lee et Lee, 2009).

XPDL offre un format standard de sérialisation pour BPMN. Plusieurs entreprises utilisent XPDL pour la définition et l'échange de processus d'affaires, dont IBM et BEA Systems.

2.2.2.4 WS-BPEL

En mai 2003, OASIS¹ a introduit le langage BPEL4WS (*Business Process Executable Language For Web Services*). BPEL4WS est un langage de modélisation de processus d'affaires dans le contexte des services web. Il permet de modéliser des processus exécutables et abstraits². BPEL4WS, maintenant WS-BPEL ou encore BPEL est issu des langages WSFL (*Web Services Flow Language*) d'IBM et XLAN (*XML LANGuage*) de Microsoft. Il a été rapidement adopté par plusieurs grandes organisations telles qu'IBM, BEA Systems, SAP et Microsoft.

WS-BPEL permet de décrire des interactions complexes. Il distingue entre deux types de processus : (i) les processus exécutables et (ii) les processus abstraits. Un processus exécutable permet de décrire les processus internes à l'entreprise. Un processus abstrait permet de décrire les processus de l'organisation qui sont exposés publiquement dans le contexte d'interactions interentreprises. Ces processus permettent de spécifier le comportement des partenaires par rapport aux messages échangés, sans rendre public le comportement interne.

WS-BPEL est un langage basé sur XML. Il utilise WSDL (*Web Services Description Language*) pour spécifier les actions du processus opérationnel et pour décrire

¹ OASIS : Organization for the Advancement of Structured Information Standards

² Processus abstrait est un processus qui spécifie le comportement mais qui n'est pas conçu pour s'exécuter en cachant le détail des ses opérations internes.

```

<message name="POMessage">
  <part name="customerInfo" type="sns:customerInfo"/>
  <part name="purchaseOrder" type="sns:purchaseOrder"/>
</message>
<message name="InvMessage">
  <part name="IVC" type="sns:Invoice"/>
</message>

```

Figure 2.2 Un exemple de messages en BPEL.

les services web offerts par le processus. La définition d'un processus avec WS-BPEL consiste en une partie déclarative des différents éléments du processus suivie par la partie qui décrit le processus. La partie déclarative permet de décrire : (i) les messages échangés entre les services, (ii) les services appelés, (iii) les interactions entre les partenaires, (iv) les partenaires, (v) les variables utilisées durant la collaboration d'affaires, et (vi) le gestionnaire d'erreurs. La Figure 2.2 montre un exemple de messages en BPEL. La structure d'un message est semblable à celle utilisée pour les services web. Elle consiste en plusieurs parties. Chaque partie a un nom et un type.

L'orchestration est définie par un ensemble d'activités partiellement ordonnées. Ces activités correspondent aux opérations définies dans les différents services. WS-BPEL supporte les activités séquentielles et les activités concurrentes. Il intègre des mécanismes de contrôle pour la gestion des flux des activités. Comme BPML, BPEL-WS supporte la gestion des exceptions et le mécanisme de compensation.

WS-BPEL met l'emphasis sur l'aspect fonctionnel et comportemental d'un processus. Il présente les concepts organisationnels de rôles et partenaires, mais n'intègre pas les acteurs humains (Mili *et al.*, 2008).

2.2.2.5 BPDM

Avec la prolifération des normes de modélisation de processus d'affaires, OMG a initié BPDM (*Business Process Definition Metamodel*) en 2003. L'objectif de BPDM est de fournir un modèle standard pour unifier l'ensemble des normes de modélisation de processus d'affaires. Le langage BPDM est indépendant de toute notation graphique et de toute

méthodologie de gestion de processus d'affaires et de *workflow* (OMG, 2008). La spécification de BPDM n'a vu le jour qu'en novembre 2008.

Le paragraphe suivant est un extrait de la spécification de BPDM (OMG, 2008) :

[...] "The meta model behind BPDM uses the OMG Meta Object Facility (MOF) standard to capture business processes in a very general way, and to provide an XML syntax for storing and transferring business process models between tools and infrastructures. Various tools, methods, and technologies can then map their way to view, understand, and implement processes to and through BPDM".

BPDM est une norme d'échange. De là, les entreprises peuvent continuer à utiliser les mêmes outils de modélisation de processus en leur possession. BPDM dans sa version 1.0 est considéré relativement immature et très complexe (Ko, Lee et Lee, 2009). La Figure 2.3 montre la position de BPDM au sein de la hiérarchie de quatre niveaux de modélisation d'OMG.

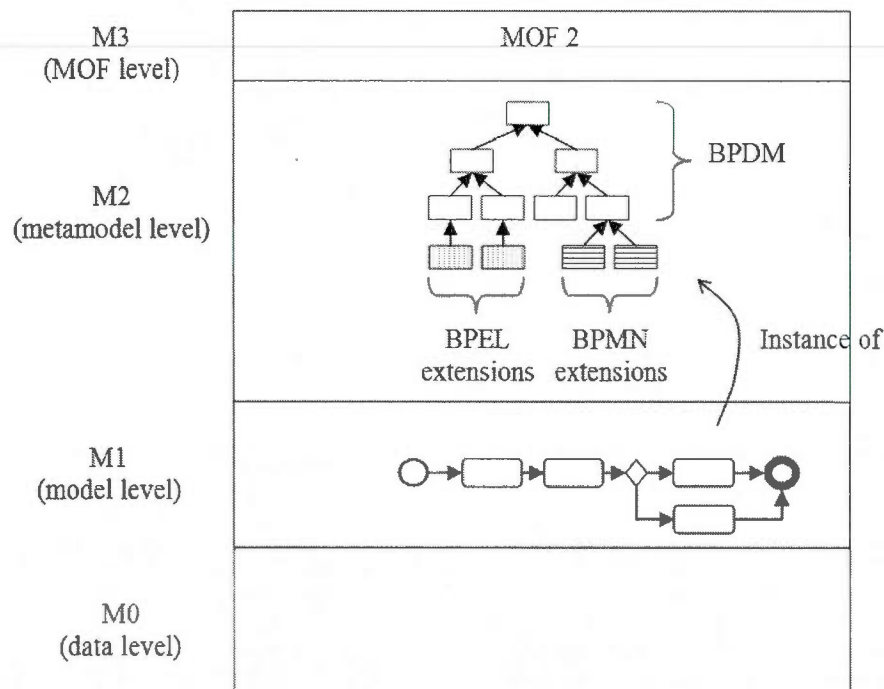


Figure 2.3 La hiérarchie de modélisation d'OMG. (Source : OMG, 2008)

2.2.3 Langages d'intégration de processus

Avec l'avènement du commerce électronique et la nécessité de collaboration entre les entreprises; plusieurs langages d'interaction interentreprises (B2B) ont vu le jour. Ces nouveaux langages de modélisation mettent l'emphasis sur les mécanismes d'intégration en termes d'indépendance technologique, d'interfaces de programmation et de formats d'échange de données entre les entreprises (Mili *et al.*, 2010). Ces langages de modélisation ont une sémantique opérationnelle qui suppose l'existence d'un moteur pour l'exécution des processus.

2.2.3.1 RosettaNet

Fondé en 1998, RosettaNet est un consortium qui regroupe plus que 1000 membres dans le secteur des technologies de l'information, des télécommunications, de la logistique, des composants électroniques et des semi-conducteurs (RosettaNet, 2010). Parmi ces membres, on retrouve Intel, Microsoft, Oracle, SAP et Cisco. Le but de ce consortium est de créer des normes pour définir un langage de processus d'affaires facilitant le commerce interentreprises. RosettaNet présente plusieurs normes permettant l'automatisation de certains processus d'affaires, particulièrement les processus d'approvisionnements et de manufactures. RosettaNet considère que les entreprises doivent assurer la compatibilité à différents niveaux de l'infrastructure informatique pour que l'échange d'affaires puisse avoir lieu.

La structure de la norme RosettaNet est formée de plusieurs couches (fig. 2.4). Elle permet aux entreprises d'être compatibles à différents niveaux. Dans chaque couche, RosettaNet définit une norme. Parmi ces normes, on retrouve le PIP (*Partner Interface Processes*), les dictionnaires et le cadre de mise en œuvre RNIF (*RosettaNet Implementation Framework*).

Les dictionnaires définissent le vocabulaire des transactions électroniques. Il existe deux types de dictionnaires : un pour les termes d'affaires (*RosettaNet Business Dictionary*) et l'autre pour les termes techniques (*RosettaNet Technical Dictionary*).

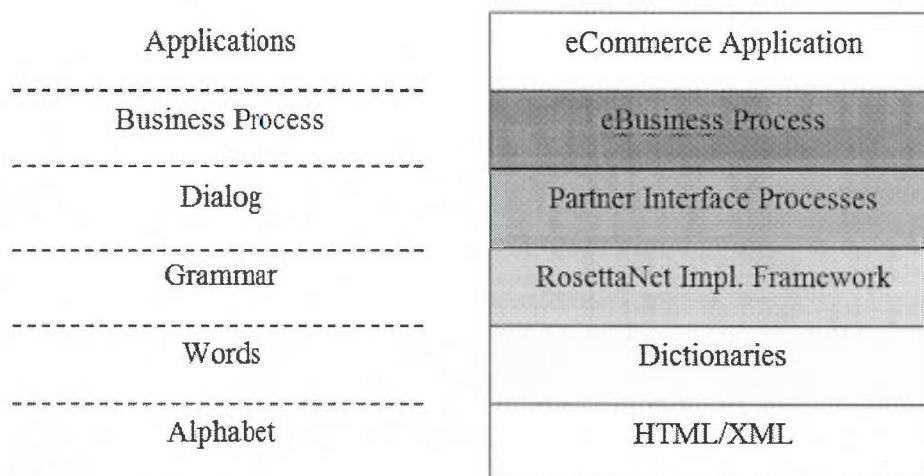


Figure 2.4 Structure de la norme RosettaNet. (Source : RosettaNet, 2010)

Les PIPs sont des modèles de processus génériques impliquant deux ou plusieurs partenaires. La spécification PIP contient trois vues du processus : (i) la vue opérationnelle d'affaires pour représenter la sémantique d'affaires, (ii) la vue du service fonctionnel qui décrit les composants de l'interaction, leurs protocoles et établit une correspondance entre les actions de PIP et les documents, et (iii) la vue de mise en œuvre du cadre RNIF qui spécifie les formats des messages. Dans la version 2010, les formats des messages sont principalement en XML DTDs (*Document Type Definitions*). Certains sont disponibles en schéma XML (XSD). La Figure 2.5 montre un exemple de processus générique d'émission de bon de commande. Ce processus provient du catalogue des PIPs publié en décembre 2011 (RosettaNet, 2011).

Le RNIF est une spécification des protocoles d'échanges pour PIP. Elle couvre : (i) les formats et les directives d'utilisation des messages d'affaires, (ii) les services de sécurité, (iii) les procédures et les règles d'assemblage et de désassemblage des messages, et (iv) les protocoles de transfert des messages et la sémantique de leur flux.

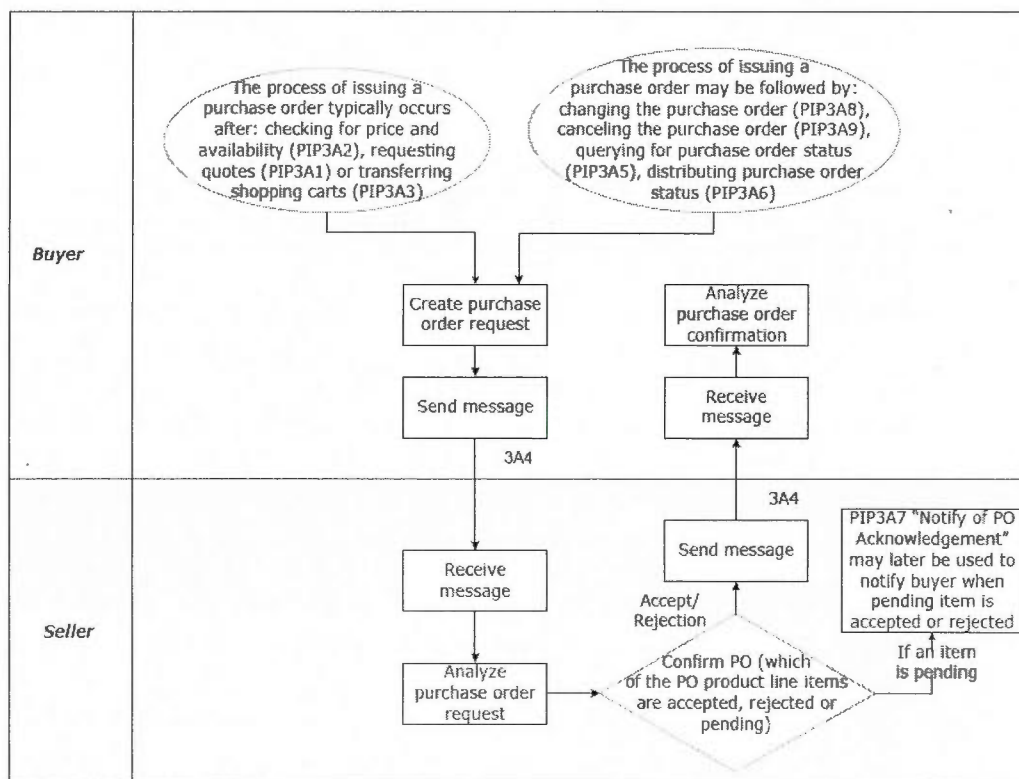


Figure 2.5 Processus d'émission de bon de commande. (Source : RosettaNet, 2011)

2.2.3.2 ebXML

En 2001, OASIS et l'UN/CEFACT³ ont formalisé le langage ebXML (*Electronic Business using eXtensible Markup Language*) afin de rendre possible aux entreprises de toutes tailles, opérant dans n'importe quel domaine, de collaborer avec toute autre entreprise dans le monde. ebXML propose de nouvelles normes d'échanges pour le commerce électronique B2B. L'échange entre les partenaires se fait par le biais de documents XML sur Internet. Contrairement à RosettaNet, ebXML est une collection de normes génériques qui ne dépendent d'aucun domaine d'affaires (Ko, Lee et Lee, 2009). Parmi les normes d'ebXML, nous citons BPSS (*Business Process Specification Schema*) et CPP (*Collaboration Protocol Profile*). La famille des normes ebXML aborde les domaines suivants (Mili *et al*, 2010):

³ UN/CEFACT: United Nations Centre for Trade Facilitation and Electronic Business

1. La norme BPSS pour décrire des processus d'affaires. Cette dernière sert à définir la collaboration entre des partenaires (B2B) dans un processus d'affaires à travers les transactions et les échanges de documents.
2. La norme CPP permet aux entreprises de représenter les processus d'affaires qu'elles supportent et comment elles les supportent.
3. Un mécanisme pour enregistrer et rechercher des CPP à travers un registre public (*ebXML Registry Services*).
4. Un mécanisme pour établir la correspondance entre deux CPPs pour produire un accord de collaboration (*Collaboration Protocol Agreement, CPA*).
5. Une librairie de processus d'affaires de base (*Core business processes*) et les documents qui couvrent les scénarios d'affaires les plus fréquents.

La Figure 2.6 montre un scénario de collaboration avec ebXML.

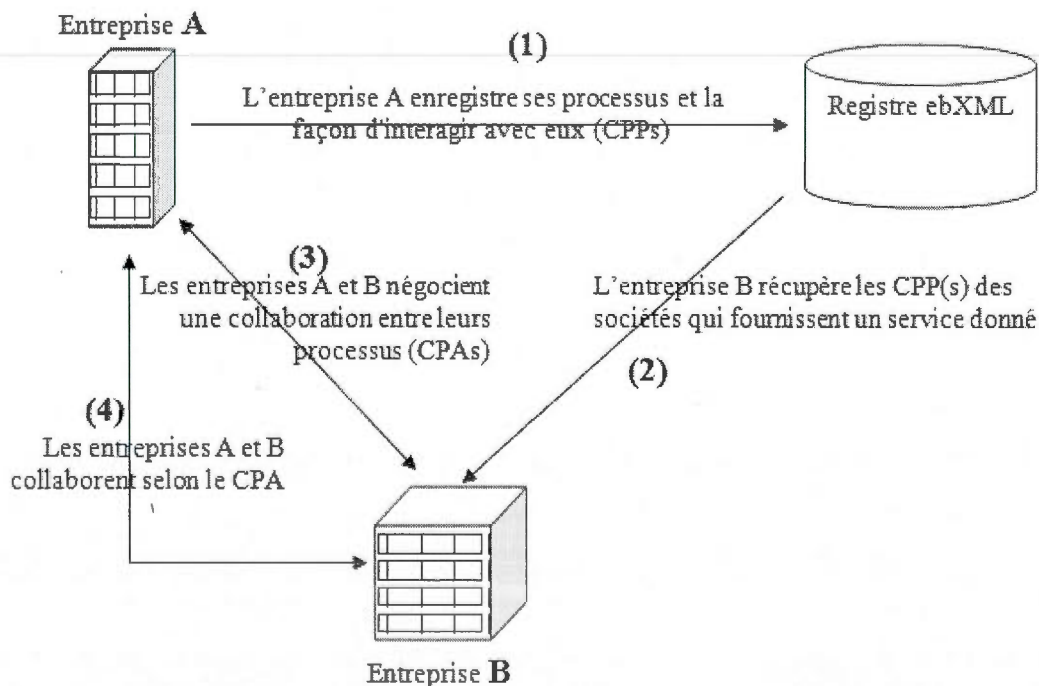


Figure 2.6 Scénario de collaboration avec ebXML. (Source : Mili *et al.*, 2010)

Avec BPSS, ebXML couvre adéquatement la vue comportementale. La vue informationnelle est partiellement traitée avec les structures des documents et les bibliothèques des composants (Mili *et al.*, 2010).

2.2.4 Langages de modélisation orientée objet

En 2004, OMG a normalisé UML2 et a introduit de nouveaux concepts d'EDOC⁴ pour la représentation des aspects comportementaux et architecturaux. Cette version est enrichie avec des notations de composition, d'agrégation comportementale et structurelle ainsi que plusieurs autres concepts de bas niveau. UML2 englobe largement la vue dynamique, la vue fonctionnelle et la vue informationnelle. Le diagramme d'activité dans UML 2.0 utilise la sémantique des réseaux de Pétri. Il représente un bon langage de modélisation comportementale avec un support pour la détection des exceptions, la gestion d'erreurs et aussi la présentation des activités composées avec la possibilité de modélisation des partitions. Néanmoins, il demeure complexe, notamment pour les analystes d'affaires, et continue de perdre du terrain chez les praticiens relativement à d'autres langages comme BPEL (Ko, Lee et Lee, 2009).

2.2.5 Les ontologies d'affaires

Nous avons présenté une variété de langages de modélisation de processus d'affaires. Ces langages font partie de quatre grandes familles: (i) les langages traditionnels, (ii) les langages dynamiques, (iii) les langages d'intégration de processus, et (4) les langages orientés objet. Nous proposons d'étudier maintenant la famille des ontologies d'affaires. Une ontologie d'affaires est toute ontologie qui décrit le concept de création, de transformation et d'échange de valeurs économiques. Parmi les ontologies d'affaires, notons : (i) REA (*Resource Event Agent*) (Geerts et McCarthy, 2000), (ii) l'ontologie e³-value (Gordijn, Akkermans et Van Vliet, 2000), et (iii) e-BMO (*e-Business Model Ontology*) (Osterwalder et Pigneur, 2002).

⁴ EDOC (Enterprise Distributed Object Computing) est un cadre de modélisation adopté par l'OMG qui permet la conception orientée objet de systèmes d'information et d'en dériver une mise en oeuvre conformément à la méthodologie MDA.

2.2.5.1 L'ontologie REA

REA a été créé par William E. McCarthy pour modéliser des systèmes d'information dans le domaine de la comptabilité (McCarthy, 1982). REA se base sur le principe suivant : Si une entreprise veut accroître la valeur totale d'une ressource sous son contrôle, elle doit généralement diminuer la valeur de certaines de ses ressources. REA perçoit un processus d'affaires en termes de transfert ou de transformation de ressources économiques. Par la suite, REA est devenu une ontologie de modélisation de processus d'affaires (Geerts et McCarthy, 2000). Les modèles REA offrent une meilleure sémantique pour la compréhension des processus d'affaires (Hruby, 2006). Le méta-modèle de base de REA est présenté à la Figure 2.7 avec la notation entité-relation de Batini, Ceri et Navathe (1992).

Les composantes principales de l'ontologie REA sont : (1) les ressources économiques, (2) les événements économiques, et (3) les agents économiques. Les *ressources économiques* sont des objets rares et utiles qui sont sous le contrôle de l'entreprise (McCarthy, 1982). Par exemple, les produits, les services, l'argent, la matière première et la main-d'œuvre. Les *événements économiques* sont des phénomènes qui changent la valeur des ressources économiques. Les ressources sont incrémentées et décrémentées dans le contexte d'un événement économique. La relation de dualité représente le lien entre les événements économiques lors d'un processus d'échange ou de transformation de ressources. L'*agent économique* est un individu ou une organisation qui peut avoir un contrôle sur les ressources économiques.

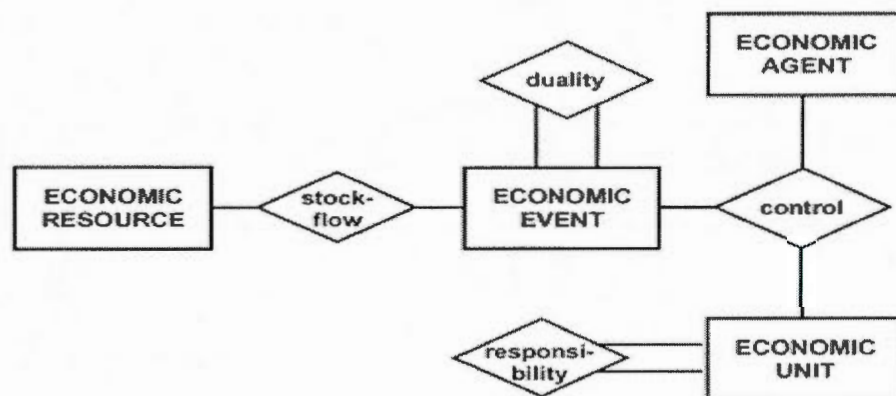


Figure 2.7 Le méta-modèle REA de base. (Source : McCarthy, 1982)

On distingue deux principaux patrons de processus d'affaires REA : le patron d'échange et le patron de transformation ou encore de conversion. Le patron d'échange consiste en un transfert de ressources économiques entre deux agents. Le patron de transformation consiste en la création de nouvelles ressources économiques ou le changement d'attributs de ressources économiques existantes par l'utilisation ou la consommation d'autres ressources. Nous présentons l'ontologie REA avec plus de détails dans le chapitre 4.

2.2.5.2 L'ontologie e³-Value

L'ontologie e³-Value a été conçue par Gordijn, Akkermans et Van Vliet (1999) dans le but d'aider à définir comment la valeur économique est créée et échangée au sein d'un réseau d'acteurs. L'objectif principal de l'ontologie est d'offrir une notation graphique formelle pour la modélisation et permettre d'évaluer le profit économique à partir du modèle. Particulièrement, elle permet de définir et analyser les relations multi-organisationnelles, les scénarios d'affaires et les exigences opérationnelles d'une manière à la fois qualitative et quantitative (Gordijn et Akkermans, 2001). L'ontologie e³-Value distingue trois vues pour décrire les modèles d'affaires : (i) la vue acteur globale (*global actor view*), (ii) la vue acteur détaillée (*detailed actor view*), et (iii) la vue des activités de valeur (*value activity view*). La *vue acteur global* montre une vue globale des acteurs impliqués et quels objets de valeur ils échangent. La *vue acteur détaillée* montre des aspects de décomposition de la vue acteur

globale (e.g. des alliances entre les acteurs). La *vue des activités de valeur* souligne les activités clés et les acteurs qui les performement.

L'ontologie e³-Value propose une représentation graphique pour chacun de ses composants. La Figure 2.8 illustre un exemple d'échange économique montrant différents concepts de l'ontologie (Schuster et Motal, 2009).

Les concepts les plus importants de l'ontologie e³-Value (Gordijn, Akkermans et Van Vliet, 2000) sont :

- A. L'acteur : Toute entité économique indépendante.
- B. L'objet de valeur (*Value object*) : C'est une ressource de valeur que les acteurs échangent. Par exemple, les produits et les services.
- C. L'échange de valeur (*Value exchange*) : Il permet de relier deux acteurs. Plus exactement, deux ports de valeur.
- D. Le port de valeur (*Value port*) : C'est un concept utilisé par un acteur pour montrer qu'il veut fournir ou acquérir des objets de valeur.
- E. L'interface de valeur (*Value interface*) : C'est un groupe de ports de valeur. Un acteur peut avoir plusieurs interfaces de valeur.

D'autres concepts tels que le « *Value offering* », « *Value activity* », « *Composite actor* » et « *Market segment* » sont présent dans Gordijn, Akkermans et Van Vliet (2000). Les scénarios d'affaires de l'ontologie e³-Value sont représentés avec la notation « *Use Case Maps* » (Buhr, 1998).

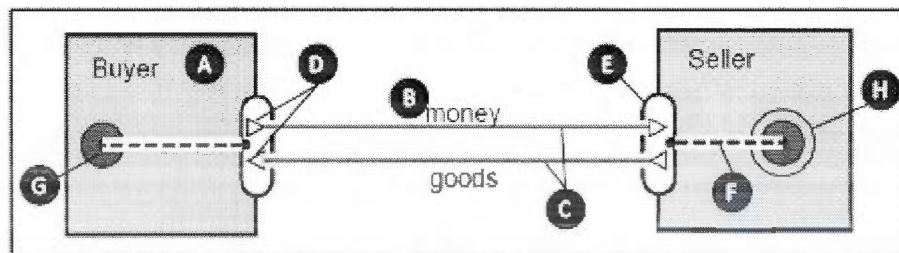


Figure 2.8 Un exemple d'échange avec l'ontologie e³-Value. (Source : Schuster et Motal, 2009)

2.2.5.3 L'ontologie e-BMO

L'ontologie e-BMO (*e-Business Model Ontology*) a été proposée par Osterwalder et Pigneur en 2002 puis finalisée par Osterwalder dans sa thèse en 2004. Osterwalder et Pigneur (2002) définissent un modèle d'affaires comme l'architecture conceptuelle de l'entreprise et son réseau de partenaires pour la création et le transfert de valeurs et de capital relationnel dans le but de générer du revenu profitable et durable.

Les composants de base de l'ontologie e-BMO sont : (i) *les produits et services* offerts par une firme qui forment une valeur essentielle pour laquelle un client est disposé à payer, (ii) *l'infrastructure et le réseau de partenaires* pour la création de valeur et le maintien d'une bonne relation client, (iii) *le capital relationnel* qui permet à l'organisation de satisfaire les demandes de ses clients et générer des revenus profitables et durables, et (iv) *les aspects financiers* tels que les coûts et les structures de revenu. La Figure 2.9 illustre les quatre composants de l'ontologie e-BMO (Osterwalder et Pigneur, 2002).

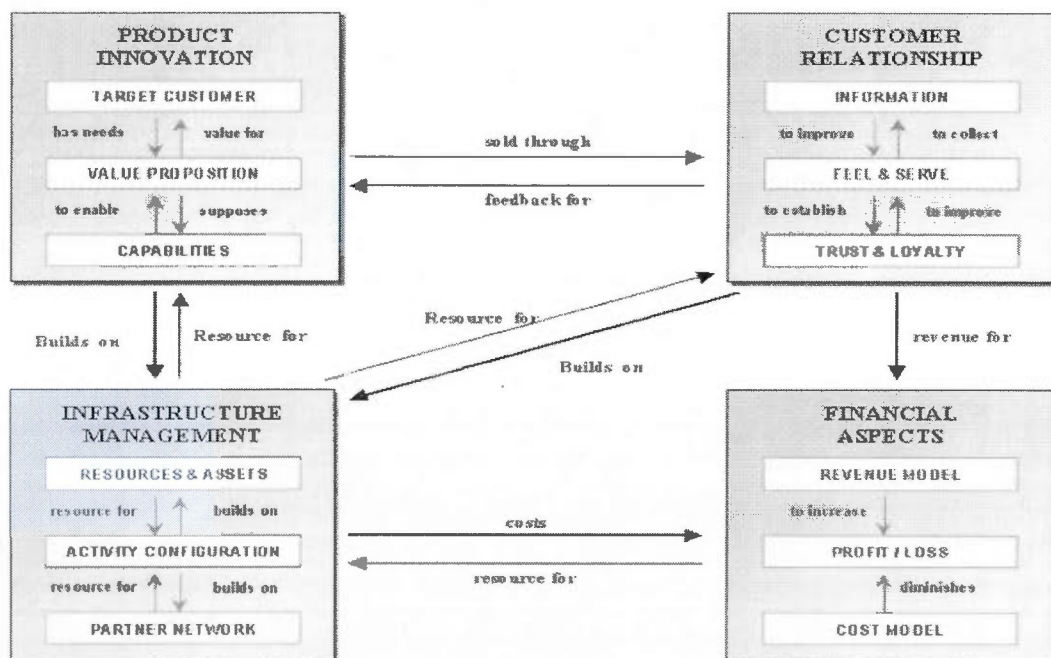


Figure 2.9 Les quatre composants de l'ontologie e-BMO. (Source : Osterwalder et Pigneur, 2002)

2.2.6 Comparaison des langages de modélisation des processus d'affaires

Le Tableau 2.1 montre, pour chaque langage de modélisation, le niveau de support de chacune des vues de processus d'affaires. « Oui » indique que le langage supporte la vue en question. « Référence » indique que le langage ne supporte pas la vue, mais que les concepteurs du langage ont lié les modèles de cette vue à des modèles dans un autre langage. Nous utilisons « En partie » si le langage présente un ensemble incomplet de structures/concepts pour représenter la vue, et « Ingrédients » si le langage contient les ingrédients mais pas encore la portée pour supporter la vue.

Tableau 2.1

Comparaison des langages de modélisation de processus (Adapté de Mili *et al.*, 2010)

Langage	Vue Informationnelle	Vue Fonctionnelle	Vue Dynamique	Vue Organisationnelle
IDEF	Oui	Oui	Oui	
RAD	En partie		Oui	En partie
EPC	Référence	En partie	Oui	Référence
BPML	Référence		Oui	En partie
XPDL	Référence		Oui	
BPMN		Oui	Oui	En partie
BPDM		Oui	Oui	Référence
EbXML	Oui		Oui	En partie
BPEL-WS	Référence		Oui	
UML2	Oui	Oui	Oui	Ingrédients
REA	Ingrédients	Oui		En partie
RosettaNet	Oui		Oui	

Le langage BPMN couvre la vue dynamique et fonctionnelle. De plus, il est largement accepté par la communauté. UML 2 englobe les vues dynamique, fonctionnelle et informationnelle. Cependant, pour la vue organisationnelle, UML 2 contient les ingrédients mais pas encore la portée. Notons aussi que le diagramme d'activité (vue dynamique) dans UML 2 demeure complexe, notamment pour les analystes d'affaires. BPDM couvre largement la vue dynamique et la vue fonctionnelle. De plus, c'est une norme récente qui spécifie les mécanismes d'échange et qui offre une sémantique bien définie des concepts d'orchestration et de chorégraphie (OMG, 2008).

Pour notre modélisation de processus d'affaires, nous allons considérer seulement les normes. Nous allons alors considérer : (i) UML (diagramme de classe) pour la vue informationnelle, (ii) BPMN ou BPDM pour les vues dynamique et fonctionnelle, et (iv) BPDM avec la norme OSM (*Organizational Structure Metamodel*) pour la vue organisationnelle.

2.3 Initiatives de réutilisation de processus d'affaires

Dans cette section, nous nous concentrons sur les initiatives de réutilisation de processus ou de besoins d'affaires. Nous montrons les initiatives à base : (i) de patrons d'analyse, (ii) de patrons d'affaires, (iii) de cadres d'application, et (iv) de méthodologies de gestion de processus d'affaires. Les patrons d'affaires et d'analyse proposent la réutilisation de modèles partiels. Les initiatives à base de cadres d'application et de méthodologies offrent une solution globale pour la réutilisation de processus d'affaires. Pour chacune de ces initiatives, nous discutons des techniques utilisées pour la spécification de processus d'affaires.

2.3.1 La réutilisation par les patrons d'analyse

Un patron d'analyse, dans l'ingénierie des systèmes, offre une solution à un problème récurrent rencontré durant l'analyse d'un domaine d'affaires. La solution est un modèle conceptuel abstrait réutilisable ou adaptable à différentes situations selon les besoins d'affaires. Plusieurs travaux ont traité ce sujet, notamment ceux de Coad, North et Mayfield (1997), de Fowler (1997) et de Wohed (2000, 2001).



Figure 2.10 Le patron « Participant-Transaction ». (Source : Coad, North et Mayfield, 1997)

Coad, North et Mayfield (1997) présentent des patrons d'analyse selon les quatre catégories suivantes : transaction, agrégation, plan et interaction. La représentation des modèles est basée sur la notation OMT (*Object Modeling Technique*) - l'un des langages précurseurs d'UML. La Figure 2.10 montre le patron *Participant-Transaction*. Un participant peut représenter un acheteur, un client, un vendeur, etc. Une transaction peut représenter par exemple un contrat, une livraison ou une commande.

Ces patrons ne sont pas liés à un domaine particulier. Ils sont simples mais difficilement applicables en raison de leur haut niveau d'abstraction. La spécialisation pour un domaine donné suit une stratégie dirigée par les questions pour assister l'analyste à identifier les attributs, les liens et les services des entités (Coad, North et Mayfield, 1997). Parmi les questions utilisées, « *what I know ?* » pour identifier les attributs, « *who I know ?* » pour identifier les liens et « *what I do ?* » pour les services.

Martin Fowler (1997) a proposé des patrons d'analyse spécifiques. Ces patrons sont classés selon leurs domaines d'affaires (e.g. la comptabilité, les inventaires, la planification, etc.). Représentés en UML, ces patrons fournissent plusieurs adaptations du même problème. Pour identifier un patron, l'utilisateur doit établir une correspondance entre le domaine d'affaires qu'il modélise et les domaines d'affaires de ces patrons d'analyse. La Figure 2.11 illustre le patron d'analyse d'une transaction comptable.

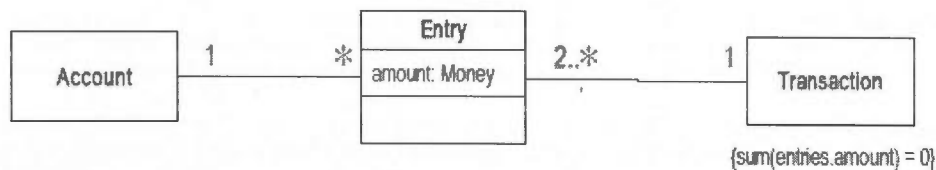


Figure 2.11 Le patron Transaction. (Source : Fowler, 1997)

Fowler (1997) ne propose pas une classification claire de ses patrons. Par conséquent, il n'existe pas de structure de navigation précise pour tous les patrons. Pour la spécialisation, l'analyste identifie un patron selon ses besoins d'affaires. Par la suite, il doit l'adapter librement (e.g. en effectuant des substitutions).

Dans un autre travail, Wohed (2000) a proposé différents modèles d'analyse dans le domaine de la réservation (voyage, hôtel, cinéma, etc.). Ces modèles, représentés en UML, étaient basés sur le patron d'analyse *Resource Allocation* de Fowler (Fowler, 1997). Pour la spécialisation, Wohed (2000) utilise une stratégie par question pour adapter les modèles d'analyse. En effet, Wohed a identifié un ensemble de questions spécifiques en analysant les différences entre plusieurs modèles de réservation. Ces questions ont été généralisées dans un deuxième travail en 2001 (Wohed, 2001) en les liant au patron *Transaction* de Coad, North et Mayfield (1997). Avec cette généralisation, les questions sont devenues indépendantes du domaine de la réservation mais elles ont perdu leur sémantique d'affaires (Wohed, 2001). Nous présentons l'ensemble des questions spécifiques et génériques de Petia Wohed (2000, 2001) dans l'appendice A.

Les transformations proposées par Wohed (2000, 2001) permettent d'adapter les modèles avec des extensions orientées objet. Nous citons par exemple, les transformations de généralisation, de spécialisation et d'ajout de super-types.

2.3.2 La réutilisation par les patrons d'affaires

Un patron d'affaires offre une description générique des règles permettant de développer une solution pour un processus d'affaires particulier. Le patron d'affaires doit décrire, dans un haut niveau d'abstraction, les objectifs de la solution, les participants, les interactions et les données (Adams *et al.*, 2001). Dans cette section, nous étudions les travaux d'Adams *et al.* (2001) et d'Eriksson et Penker (2000).

Dans le projet de développement des patrons d'affaires en commerce électronique d'IBM (*Patterns for e-business*), Adams *et al.* (2001) proposent des patrons d'affaires ainsi que des patrons de spécialisation. Ces patrons d'affaires, devenus *Patterns for e-business(P4eb)* en 2010, modélisent différents concepts de processus d'affaires. Adams *et al.* (2009) distinguent les patrons suivants : *Libre-service, collaboration, agrégation*

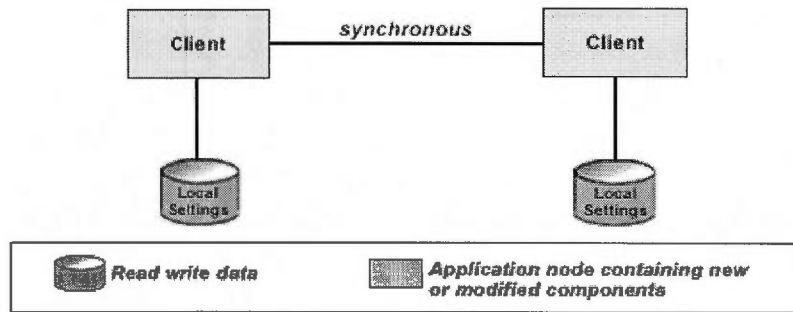


Figure 2.12 Le patron d'application « point-à-point ». (Source : Adams *et al.*, 2009)

d'information et entreprise étendue. Le patron *Libre-service* modélise les interactions des utilisateurs avec les organisations (*User-to-Business*). Le patron *Collaboration* modélise les interactions entre les utilisateurs (*User-to-User*). Le patron *Agrégation d'information* modélise les interactions entre les utilisateurs et les données (*User-to-Data*). Finalement, le patron *Entreprise étendue* capture les interactions entre les organisations (*Business-to-Business*). Adams *et al.* (2001) proposent d'autres patrons d'intégration et de composition pour combiner les patrons d'affaires afin de construire un modèle complet. Pour spécialiser un patron, il faut naviguer un catalogue de modèles, dit référentiel de patrons d'applications, qui représente les différentes spécialisations d'un patron d'affaires. Par exemple, le patron d'application *liaison point-à-point* est une spécialisation du patron d'affaires *Collaboration* (fig. 2.12). Il permet à deux intervenants de communiquer d'une manière synchrone. Une fois le patron d'application sélectionné, Adams *et al.* (2001) offrent à l'analyste des patrons d'exécution. Finalement, la dernière étape consiste à choisir des correspondances avec des composants logiciels pour la mise en œuvre.

Eriksson et Penker (2000) ont proposé un catalogue de patrons d'affaires comportementaux et structurels. Ces patrons abordent les mêmes domaines d'affaires que Fowler (1997) à l'exception du domaine d'analyse financière. La représentation de ces patrons utilise le diagramme de classe UML pour la structure et le diagramme d'activités avec quelques extensions IDEF (sect. 2.2) pour la vue comportementale. La Figure 2.13 montre le patron *Structure d'un contrat* qui modélise les concepts de base d'un contrat commercial.

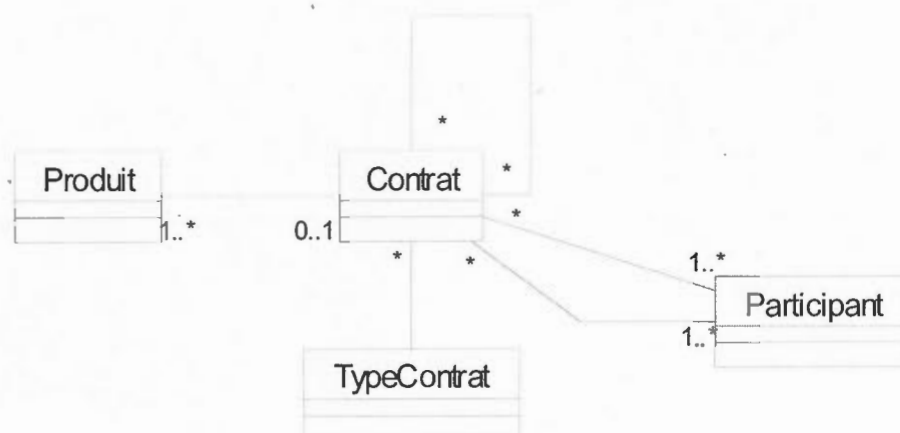


Figure 2.13 Le patron « Contrat ». (Source : Eriksson et Penker, 2000)

Eriksson et Penker (2000) distinguent trois classes de patrons d'affaires : (i) les patrons de ressources et de règles, (ii) les patrons de processus, et (iii) les patrons d'objectifs. Les patrons de ressources et de règles modélisent les personnes, le matériel, l'information et les produits. Nous citons, par exemples, les patrons *Acteur-rôle*, *Définitions d'affaires* et *Localisation géographique*. Les patrons de processus décrivent l'aspect fonctionnel et comportemental. Par exemple, le patron *Structure de base d'un processus* et le patron *Interaction*. Finalement, les patrons d'objectifs modélisent les cibles d'affaires tels que le patron *Décomposition des objectifs*, le patron *Problème-objectif* et le patron *contrat* qui modélise les concepts de base d'un contrat commercial (fig. 2.13).

Quant à leur application, Eriksson et Penker (2000) ne proposent pas de mécanisme de spécialisation. La technique utilisée consiste à sélectionner un patron d'affaires et l'instancier.

2.3.3 La réutilisation par les cadres d'application

Au lieu de se limiter à une modélisation de fragments de processus, différentes recherches illustrent le besoin d'avoir un cadre global de gestion de processus d'affaires. Ces cadres offrent des approches de modélisation et d'adaptation. Ils sont généralement supportés par des outils.

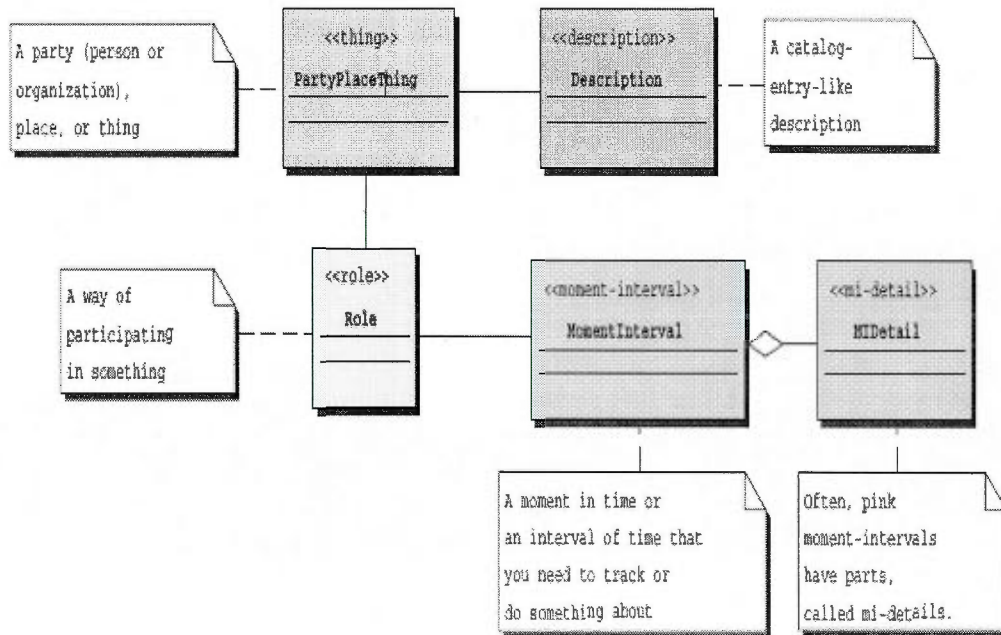


Figure 2.14 Les archétypes et leurs couleurs. (Source : Coad, Lefebvre et De Lucca, 1999)

Parmi ces travaux de recherche, Coad, Lefebvre et De Lucca (1999) proposent un cadre d'application avec des modèles génériques de composants d'affaires. Ces modèles peuvent être étendus ou modifiés. Les modèles sont représentés en UML. Le cadre préconise différents archétypes pour différents concepts et leur associe des couleurs pour en faciliter la compréhension. Coad, Lefebvre et De Lucca (1999) proposent quatre archétypes : (i) *moment-intervalle* (Rose), (ii) *rôle* (Jaune), (iii) *description* (Bleu), et (iv) *participant/place/chose* (Vert). La Figure 2.14 montre les différents archétypes.

L'archétype *Moment-Intervalle* représente les concepts qui dépendent du temps (e.g. vente, location, etc.). Ce dernier est défini comme suit (Coad, Lefebvre et De Lucca, 1999): "Something that one needs to work with and track for business and legal reasons, something that occurs at a moment in time or over an interval of time".

L'archétype *rôle* représente un rôle joué par une entité. Il est associé au moment/intervalle (fig. 2.14). L'archétype *partie/place/chose* désigne un réalisateur de rôle (e.g. une organisation, un participant, un employé, un point de vente, etc.). L'archétype

description représente une collection ou un groupe de valeurs qui permet de décrire des concepts. Cette description permet par exemple d'identifier une entrée dans un catalogue.

Coad, Lefebvre et De Lucca (1999) proposent un composant indépendant des domaines DNC (*Domain-Neutral Component*) et des composants du catalogue plus élaborés. Le DNC est construit à partir des concepts qui représentent un processus d'affaires. Il est composé des archétypes colorés reliés entre eux. Le DNC peut être appliqué et spécialisé dans différents domaines d'affaires pour modéliser les composants d'affaires. Pour cela, les auteurs préconisent la technique suivante : (1) identifier les étapes du processus, (2) identifier les éléments du domaine (i.e. les acteurs, les événements et les ressources) pour chaque étape, et enfin, (3) il faut faire une correspondance entre ces éléments et les éléments du DNC selon des directives basées sur une approche par question. Cette stratégie par les questions permet de déterminer les archétypes. Par exemple, si nous prenons un processus de vente, nous identifions les concepts suivants : clients, ventes, produits et catalogue. Par la suite, nous attribuons à chacun de ces concepts l'archétype correspondant du DNC. Nous obtenons le modèle de la Figure 2.15.

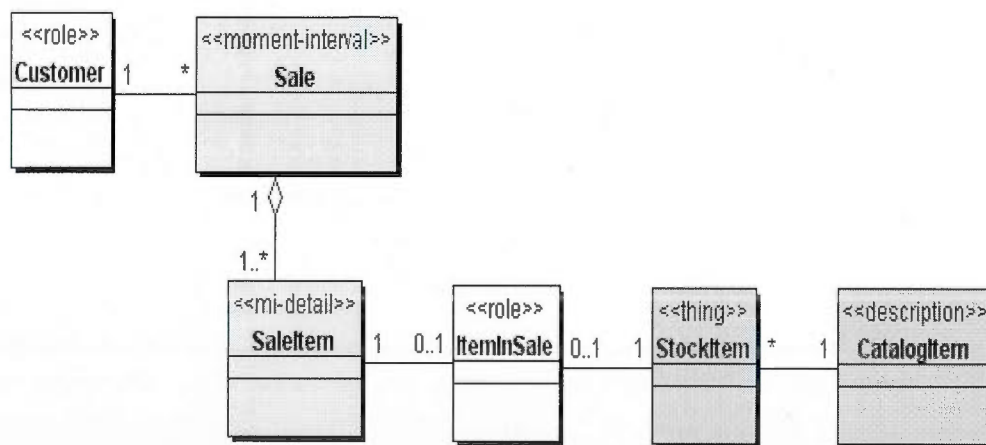


Figure 2.15 Les composants du processus de vente. (Source : Coad, Lefebvre et De Lucca, 1999)

Coad, Lefebvre et De Lucca (1999) présentent des centaines de composants spécifiques aux domaines qui suivent le modèle du DNC. Ces composants peuvent être utilisés tels quels. Ils peuvent également être spécialisés en ajoutant de nouvelles fonctionnalités à travers des points d'extension ou en éliminant celles qui ne sont pas désirées.

Curran, Keller et Ladd (1998), quant à eux, décrivent le cadre de conception et d'implémentation du modèle de processus basé sur les modèles de référence (*blueprints*) du progiciel SAP. Ces modèles sont des processus d'affaires complets qui facilitent le développement de systèmes. Le cadre d'application SAP implémente une architecture à trois niveaux : une couche présentation, une couche d'application et une couche de persistance. Le cadre SAP offre des processus génériques couvrant des domaines tels que la finance, la logistique, la gestion et d'autres domaines, dits sectoriels, spécifiques à des domaines métiers d'entreprise. Les processus SAP sont basés sur les concepts de scénarios, de processus et de transactions. Le scénario représente le processus à un haut niveau, la transaction décrit l'activité et son contexte de réalisation. La représentation des différents modèles se base sur EPC (sect. 2.2.1). La spécialisation suit une approche dite « *configure to order* ». D'abord, les modèles SAP à utiliser sont identifiés selon les besoins de l'organisation. Ces derniers peuvent être utilisés tels quels ou bien modifiés avec une gamme d'outils fournis par SAP.

Dans un travail parallèle, Rubin, Christ et Bohrer (1998) présentent le cadre IBM San Francisco comme un nouveau paradigme pour la construction des systèmes d'information. Entièrement écrit en Java, il a été intégré dans la suite *IBM Business components for WebSphere*. Ce cadre fournit essentiellement des composants d'affaires qui constituent la base de développement rapide des applications. Ce cadre est composé de trois couches (fig. 2.16). Le noyau forme l'infrastructure de base. La couche d'objets d'affaires CBO (*Common Business Objects*) regroupe différents objets de gestion et de service communs à plusieurs domaines. La couche des processus d'affaires CBP (*Core Business Processes*) propose quelques processus spécifiques à certains domaines, notamment la finance et la gestion des commandes. La spécialisation se base sur des techniques d'adaptation fondées principalement sur les concepts orientés objet. Voici quelques techniques d'adaptation :

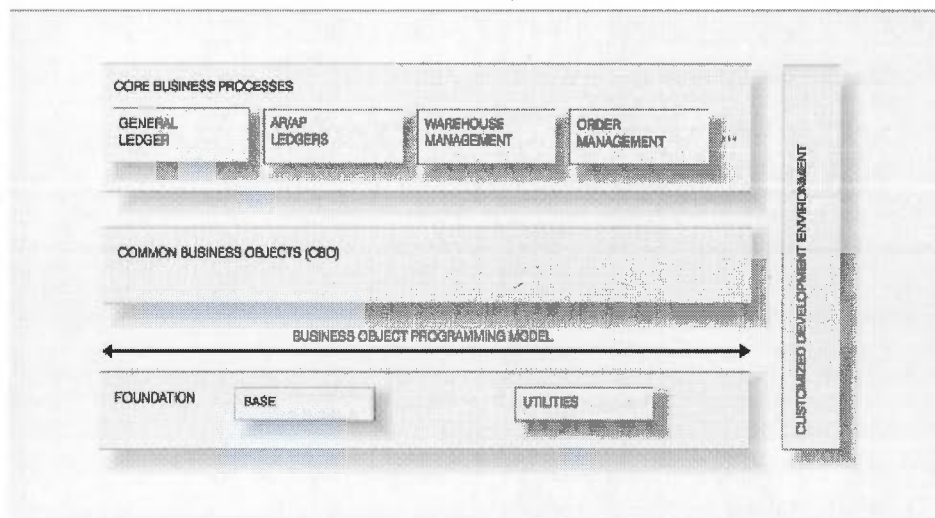


Figure 2.16 Le cadre IBM San Francisco. (Source : Rubin, Christ et Bohrer, 1998)

- Si la fonctionnalité existe mais devra être changée, le cadre fournit des points d'extension au niveau du CBP.
- Si la fonctionnalité est absente, les auteurs préconisent l'utilisation de la couche CBO ou bien en créant de nouvelles classes à partir de la couche fondation.

IBM joue un rôle important dans le domaine d'automatisation et d'optimisation de processus d'affaires. À l'heure actuelle, IBM offre une suite complète de gestion de processus d'affaires (*Business Process Management, BPM*) à travers *IBM Business Process Manager Advanced*.

Dans un travail récent, Hallerbach, Bauer et Reichert (2008) ont proposé le cadre Provop (*PROcess Variants by Options*) pour la modélisation et la gestion de spécialisation des modèles de processus d'affaires. L'approche permet de produire plusieurs variantes à partir d'un processus d'affaires unique appelé le processus de base. Provop utilise la notion d'option pour spécifier les points de variabilité. Ces options sont configurées au niveau du processus de base. Ce dernier est le processus d'affaires générique qui représente la racine de toutes les variantes au niveau de la hiérarchie. Une option est un point de variation

réutilisable qui consiste en un regroupement d'opérations de spécialisation. Les options peuvent être combinées ensemble.

Dans un deuxième travail, Hallerbach, Bauer et Reichert (2010) ont étendu le cadre Provop pour supporter une collection plus large de processus d'affaires. De plus, les auteurs proposent des stratégies pour :

- (i) modéliser le processus de base,
- (ii) organiser les variantes pour réduire l'effort global de modélisation et de maintenance,
- (iii) supporter l'évolution du processus de base tout en maintenant l'ensemble des variantes valides, et
- (iv) intégrer les concepts Provop dans des outils de modélisation de processus d'affaires.

Pour adapter un processus d'affaires, Provop supporte les opérations d'ajout, de suppression et de déplacement. La Figure 2.17 illustre un exemple de processus de base dans le domaine de la réparation automobile, décrit avec la notation EPC.

La Figure 2.18 montre trois options de spécialisation. La première option permet de modifier des activités (i.e. diagnostic et réparation). La deuxième option permet de supprimer une partie du processus d'affaires. Finalement, l'option 3 permet d'ajouter une activité entre deux points. Tous ces points de variation sont définis dans le processus de base.

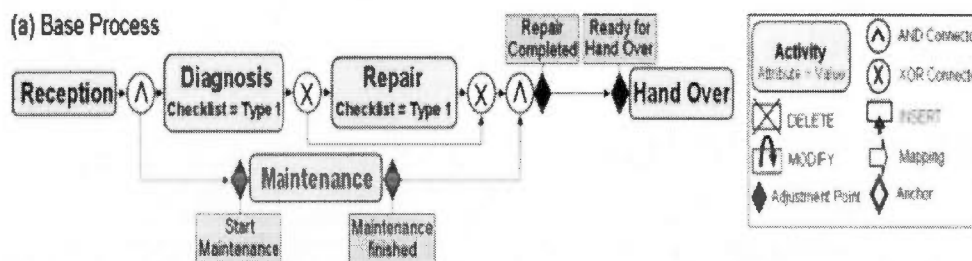


Figure 2.17 Exemple de processus de base. (Source : Hallerbach, Bauer et Reichert, 2010)

(b) Options

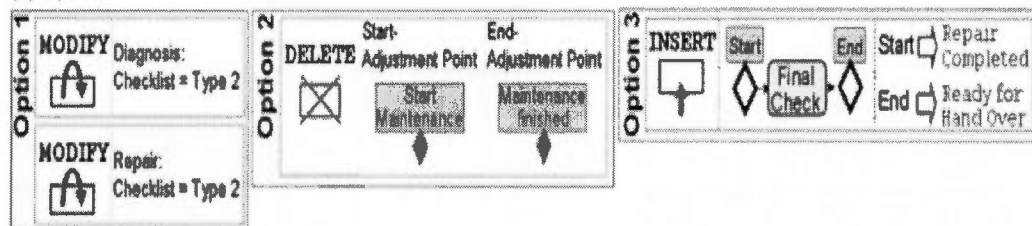


Figure 2.18 Exemple d'options du cadre Provoq. (Source : Hallerbach, Bauer et Reichert, 2010)

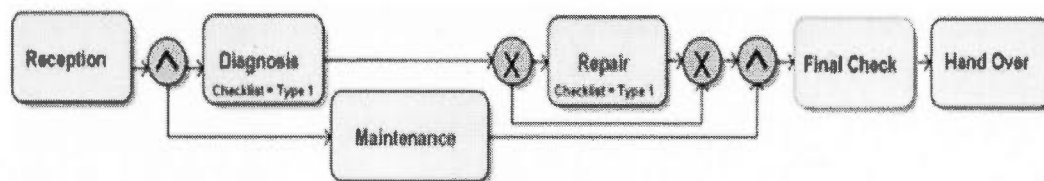


Figure 2.19 Le processus obtenu après l'application de l'option 3. (Source : Hallerbach, Bauer et Reichert, 2010)

La Figure 2.19 montre le nouveau processus d'affaires obtenu après l'application de l'option 3 (fig. 2.18) au processus de base (fig. 2.17). La spécialisation consiste à ajouter l'activité « *Final Check* » entre les deux points de variation « *Repair Completed* » et « *Ready for Hand Over* » du processus de base.

Bien que l'approche utilisée au sein du cadre Provop semble intéressante, elle présente certains inconvénients. D'abord, les options sont spécifiques à chaque famille de processus d'affaires. De plus, le concepteur a la lourde tâche de définir les contraintes sémantiques entre les options afin d'éviter les variantes non valides. Par exemple, il doit définir les règles d'implication, d'exclusion et l'ordre entre les options. De plus, le cadre ne supporte pas une hiérarchie de spécialisation. Cette dernière permet de naviguer et de localiser facilement une variante du processus et la réutiliser.

2.3.4 La réutilisation par l'approche BPM

BPM (*Business Process Management*) est une méthodologie de gestion de processus d'affaires qui remonte au début des années 90 avec l'émergence de la réingénierie des processus d'affaires (Ko, 2009). BPM est une approche systémique dont l'objectif est : (i) d'améliorer la qualité et la productivité à travers une démarche orientée processus, (ii) de permettre une gestion efficace des changements au niveau des processus d'affaires, (iii) d'offrir une vision globale des processus de l'entreprise qui permet de les optimiser et les automatiser, et (iv) d'offrir une meilleure définition des fonctions et des rôles (Ko, 2009).

Le cycle de vie d'un processus d'affaires dans le cadre de BPM suit un processus itératif. Les activités d'un cycle BPM sont : (1) la conception et la modélisation, (2) l'implémentation et l'exécution du processus, (3) la gestion et la supervision, et (4) l'optimisation. Il arrive parfois que l'optimisation ne donne pas les résultats escomptés. Dans de telles circonstances, l'approche recommande une réingénierie de l'ensemble des processus en utilisant l'approche BPR (*Business process reengineering*).

La suite BPM (BPMS) est un ensemble d'outil destiné à supporter la démarche BPM. Cet ensemble comprend : (i) un catalogue de processus d'affaires, (ii) un outil de modélisation graphique (e.g. BPMN), (iii) un outil d'aide à l'implémentation, (iv) un moteur d'exécution à base de flux (*workflow engine*) (e.g. BPEL), (v) un moteur de règle d'affaires pour l'adaptation (i.e. agilité), et (vi) des outils de pilotage et de création de rapports. La Figure 2.20 illustre les différentes normes en relation avec le cycle de vie d'un processus d'affaires dans le cadre de BPM.

L'approche BPM a eu un succès au niveau de plusieurs organisations. Un nombre important d'outils existent pour supporter cette démarche, comme *Provision* de Proforma, *Business Process Manager Advanced* d'IBM, *ARIS Design Platform*, *Business Studio* de Tibco et *System Architect* de Telelogic. Néanmoins, l'écart entre le niveau affaires et la technologie de l'information demeure important au sein de l'approche BPM (Ciuksys et Caplinskaskas, 2007).

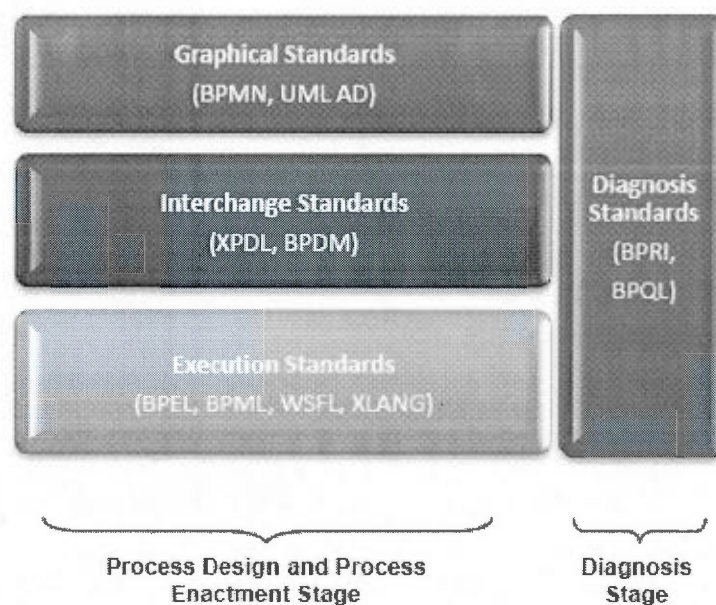


Figure 2.20 Les catégories des normes en relation avec le cycle de vie d'un processus d'affaires dans le cadre de BPM. (Source : Ko, 2009)

Dans le début de l'année 2005, l'approche SBPM (*Semantic Business Process Management*) a vu le jour. Cette nouvelle approche, qui combine la sémantique à la méthodologie BPM, vise à améliorer l'approche BPM en minimisant l'écart entre le processus d'affaires et les systèmes d'information. Le but fondamental de SBPM consiste à représenter à la fois la perspective d'affaires et la perspective des systèmes d'information avec un ensemble d'ontologies et d'utiliser un raisonnement machine pour réaliser la traduction entre les deux sphères (Hepp et Roman, 2007). SBPM vise aussi une représentation unifiée du processus d'affaires des organisations au niveau sémantique. Ceci facilite la construction de requêtes intelligentes, particulièrement, pour la découverte de processus, de fragments de processus et leur composition (Hepp et Roman, 2007). Parmi les projets de recherche SBPM, nous citons SUPER (*Semantics Utilized for Process Management within and between Enterprises*) (ESSI, 2006). SUPER est dirigé principalement par SAP avec IBM et IDS comme partenaires. La Figure 2.21 illustre les principes de l'approche SBPM.

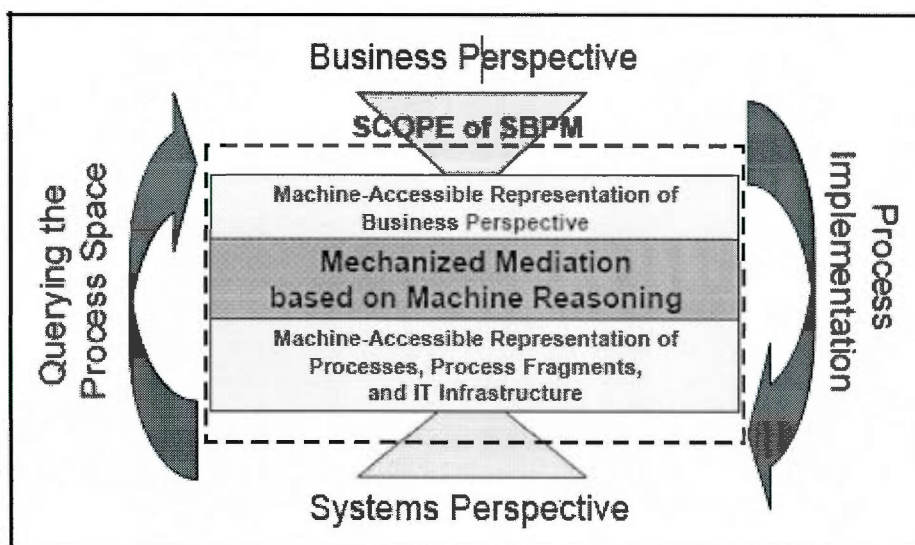


Figure 2.21 L'approche SBPM. (Source : Hepp et Roman, 2007)

Dans sa version traditionnelle, l'approche BPM n'a pas la capacité de gérer facilement les changements de processus d'affaires. La gestion des variabilités implique beaucoup d'efforts, réduit le partage et la réutilisation (IBM, 2009). Dans sa version dynamique (*Dynamic BPM*), une nouvelle couche d'agilité a été ajoutée dans le but d'adapter le processus aux changements d'une manière rapide et plus efficace. Malheureusement, aucune spécification n'existe pour la gestion de spécialisation. Chaque compagnie a adopté sa propre stratégie pour la réalisation de cet objectif.

Hallerbach, Bauer et Reichert (2010) ont étudié plusieurs outils BPM. Ils ont constaté que les processus d'affaires d'une même famille sont souvent définis et maintenus dans des modèles de processus séparés sans liens entre eux. Cette stratégie de modèles multiples présente deux inconvénients majeurs. D'abord, il en résulte des modèles redondants faiblement couplés (e.g. couplage par des conventions de nommage seulement). Ensuite, l'effort de modélisation et de maintenance est considérable quand le processus de base pour une même famille change (e.g. en raison de nouveaux règlements juridiques) (Hallerbach, Bauer et Reichert (2010).

2.3.5 La réutilisation à base de répertoriage de processus d'affaires

Publié depuis 1991, le « *MIT process handbook* » est un manuel créé par le centre de coordination et le centre du commerce électronique du MIT (*Massachusetts Institute of Technology*). Il offre une librairie riche pour le partage et la gestion des connaissances concernant les domaines et les processus d'affaires (Malone, Crowston et Herman, 2003). Les organisations peuvent consulter ce manuel pour trouver différentes alternatives ainsi que les lignes directrices qui décrivent les processus d'affaires. Cependant, ce manuel n'offre pas de technologie qui permet d'implémenter les processus. Il met l'accent sur la classification sans offrir un langage de modélisation. De plus, il n'offre pas de support informatique.

Le manuel MIT (Malone *et al.*, 1999) propose des modèles génériques de processus d'affaires de différents domaines d'affaires ainsi que des modèles spécifiques. La navigation du manuel va du modèle générique au spécifique. Elle est réalisée selon deux dimensions avec une boussole de processus (fig. 2.22). La dimension verticale distingue les différentes composantes d'un processus alors que la dimension horizontale distingue les types du processus. La classification des processus d'affaires du manuel MIT utilise une approche de hiérarchisation. La représentation des processus se base sur une explication textuelle restreinte, aucun modèle graphique ou formel n'est fourni. L'organisation des modèles des processus d'affaires se base sur trois vues : la vue à base de fonctionnalités d'affaires (e.g. logistique, marketing, etc.), la vue à base d'activités (e.g. détruire, créer, modifier, etc.) et la vue par modèles d'affaires. La spécialisation est basée sur le concept d'héritage de fonctionnalités et sur le concept de dépendance de *la théorie de la coordination*.

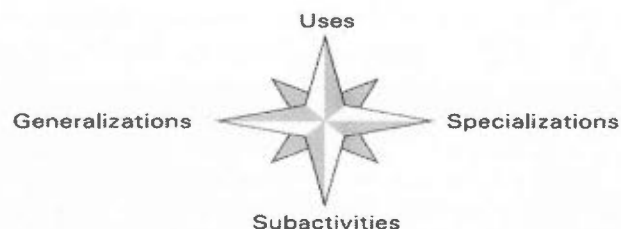


Figure 2.22 Boussole de processus du manuel MIT. (Source : Malone, Crowston et Herman, 2003)

Malone *et al.* (1999) exposent le concept de la coordination en se basant sur le principe qu'un processus est un ensemble d'activités et que les dépendances entre les activités peuvent être classifiées. La théorie de la coordination suggère des processus de coordination qui permettent de gérer la dépendance entre les activités. Dès lors, au lieu de lister explicitement toutes les activités de coordination pour différents processus, nous indiquons simplement que « la dépendance entre les activités A1 et A2 est gérée par une instance de processus de coordination P ». La Figure 2.23 montre que la navigation ainsi que la spécialisation est basée sur une stratégie dirigée par les questions.

Specializations

- ▣ Sell how?
 - ▣ Sell via store
 - Sell via face-to-face sales
 - ▣ Sell via other direct marketing
- ▣ Sell what?
 - ▣ Sell process
 - ▣ Sell service
 - Sell product
- ▣ Sell to whom?
 - ▣ Sell to businesses
 - Sell to consumers
- ▣ Sell with what customization?
 - ▣ Sell custom item to order
 - Sell standard item to order
 - Sell standard item from stock
- ▣ Sell via what channel?
 - ▣ Sell direct
 - Sell via broker
 - Sell via distributor
- ▣ Sell - views
 - Marketing- Kotler
 - ▣ Deliver execution {SCOR Dx}
 - Deliver {SCOR}
 - Deliver {SCOR*}
 - Market and Sell Products and Services {PCF 3.0}

Figure 2.23 Exemple de spécialisation au sein du manuel MIT. (Source : Malone *et al.*, 1999)

2.4 L'approche de spécialisation par les questions

Nous visons une approche basée sur les questions pour la spécification de processus d'affaires. Ce sont ces questions qui nous permettront d'identifier les transformations à appliquer aux modèles de processus d'affaires. Cette approche n'est pas nouvelle. Carlson (1979) a proposé sept questions en se basant sur le concept de commande générique : chaque entreprise fournit des biens ou des services à un client. Ces derniers sont produits suite à une commande qui peut s'appliquer à des choses, à des compétences ou à des espaces. Voici deux exemples de questions (Carlson, 1979) :

1. Le fournisseur envoie-t-il une facture au client ou celui-ci paie-t-il sa transaction comptant ou l'équivalent ?
2. Les prix sont-ils négociés (i.e. différents d'un client à l'autre) ou sont-ils fixes peu importe le client ?

Chaque réponse spécialise le processus. Par exemple, dans la question 1, si le fournisseur envoie une facture, alors il faut établir un processus de facturation; le système doit aussi conserver les informations sur les clients. Si le client paye comptant, alors le système ne requiert pas de conserver l'information sur ce dernier (Mili *et al.*, 2009).

La spécialisation dans le cadre du travail de Coad, North et Mayfield (1997) (sect. 2.3.1) suit une stratégie dirigée par les questions pour assister l'analyste à identifier les attributs, les liens et les services des entités du modèle.

Wohed (2000) a aussi proposé un nombre prédéfini de questions pour spécialiser des modèles d'analyse dans le domaine de la réservation (voyage, hôtel, cinéma, etc.). Dans ce travail, Wohed a identifié un ensemble de questions en analysant les différences entre plusieurs modèles de réservation. Voici deux questions spécifiques (Wohed, 2000):

- 1 *Est-ce que la réservation consiste en un ou plusieurs objets?*
- 2 *Est-ce que la réservation concerne un objet spécifique ou les caractéristiques d'un objet?*

Ces questions ont été généralisées dans un deuxième travail (Wohed, 2001) en les liant au patron *Transaction* de Coad, North et Mayfield (1997) (sect. 2.3.1). Voici la généralisation des deux questions précédentes (Wohed, 2001) :

- 1 *Est-ce que la transaction consiste en un ou plusieurs objets?*
- 2 *Est-ce que la transaction concerne un objet spécifique ou les caractéristiques d'un objet?*

Coad, Lefebvre et De Lucca (1999) utilisent eux aussi des questions pour identifier les différents archétypes. Tout d'abord, ils identifient les composants du domaine d'affaires. Par la suite, ils appliquent une stratégie par question pour associer ces composants aux différents archétypes du DNC ou d'un modèle adaptable choisis à partir d'un catalogue. Finalement, le manuel MIT (Malone *et al.*, 1999) utilise aussi une stratégie dirigée par les questions pour la spécialisation.

2.5 Conclusion

Nous avons montré dans ce chapitre qu'un processus d'affaires se compose essentiellement de quatre vues selon Curtis, Kellner et Over (1992): la vue fonctionnelle, la vue dynamique, la vue informationnelle et la vue organisationnelle. Nous avons aussi étudié plusieurs langages de modélisation de processus d'affaires. Ces langages proviennent de différentes traditions. Ils sont en évolution continue et couvrent différentes facettes d'un processus d'affaires. Parmi ces langages de modélisation, nous distinguons les normes suivantes : (1) les normes graphiques BPMN et le diagramme d'activité UML, (2) les normes d'échange XPDL et BPDML, et (3) les normes d'exécution BPEL et BPML.

Nous avons aussi présenté dans ce chapitre plusieurs initiatives de réutilisation de processus et de besoins d'affaires. Nous avons discuté principalement des initiatives à base de patrons d'analyse, de patrons d'affaires, de cadres logiciels et de la méthodologie de gestion BPM. Les patrons d'analyse de Coad, North et Mayfield (1997) présentent un inconvénient quant à leur application relativement à leur généralité. Les patrons d'analyse de Fowler (1997) sont spécifiques à certains domaines d'affaires seulement. Le manuel MIT (Malone *et al.*, 1999) met l'accent sur une classification textuelle sans offrir de langage de

modélisation ni de spécialisation structurelle. Les patrons d'affaires d'Adams *et al.* (2001) sont fragmentaires. De plus, ils proposent une spécialisation basée sur une navigation d'un référentiel de patrons d'application (i.e. de spécialisation) sans classification structurelle. Les patrons d'affaires d'Eriksson et Penker (2000) sont spécifiques à certains domaines. Dans le cadre de SAP, l'adaptation des processus d'affaires n'est pas claire (Rosemann et van der Aalst, 2003). En plus, à l'instar des systèmes PGI, il arrive souvent que l'organisation s'adapte aux exigences des processus d'affaires et non le contraire (Ciuksys et Caplinskas, 2007). L'approche proposée par Coad, Lefebvre et De Lucca (1999) peut être complexe pour un analyste d'affaires car elle exige une bonne connaissance du langage UML. Le cadre IBM San Fransisco souffre d'un écart important entre la sémantique des composants logiciels et les processus d'affaires qu'ils soutiennent. De plus, il représente seulement 40 % des efforts de développement. L'approche BPM présente un inconvénient quant à l'écart important entre le processus d'affaires et le système d'information (Ciuksys et Caplinskas, 2007). Cela rend difficile la réutilisation et la spécialisation de processus d'affaires par un analyste non technique. Finalement, les options du cadre Provop (Hallerbach, Bauer et Reichert, 2008) sont spécifiques aux familles de processus d'affaires. De plus, l'approche exige la connaissance préalable des points de variations et n'offre pas de hiérarchie de spécialisation.

Nous avons discuté dans ce chapitre de l'approche de spécialisations de processus d'affaires par les questions. Nous avons présenté quelques initiatives de spécialisation (Carlson, 1979; Coad, North et Mayfield, 1997; Coad, Lefebvre et De Lucca, 1999; Malone *et al.*, 1999; Wohed, 2000). Ces initiatives ne sont pas complètes. De part, les questions ne s'appliquent pas à tous les domaines, ou bien, elles présentent l'inconvénient d'être spécifiques à des processus particuliers. Lorsque les questions de Wohed (2000) ont été généralisées avec le patron *Transaction* de Coad, North et Mayfield (1997), la sémantique d'affaires était perdue (Wohed, 2001).

Dans le cadre de ce travail, nous présenterons une nouvelle approche de spécialisation automatique de processus d'affaires. Notre approche est basée sur des questions et des transformations génériques qui peuvent s'appliquer à des processus indépendamment de leur domaine d'affaires.

CHAPITRE III

VUE GLOBALE DE NOTRE APPROCHE DE SPÉCIFICATION DE PROCESSUS D'AFFAIRES

Nous avons présenté, dans le chapitre 2, différentes initiatives de spécification de processus d'affaires. Nous avons aussi discuté de leurs limites. Principalement, ces initiatives sont spécifiques à certains domaines d'affaires ou difficiles à appliquer.

Notre recherche a pour objectif de développer une nouvelle méthodologie et des outils pour la réutilisation et l'adaptation d'un catalogue de processus génériques. Nous visons une approche de spécification de processus d'affaires facile à assimiler par des utilisateurs ayant peu de connaissances techniques. Notre technique d'adaptation de processus d'affaires est dirigée par les questions (sect. 2.4). Elle préconise le passage d'un modèle générique de processus d'affaires à un modèle spécifique à l'organisation. Le processus d'affaires générique est choisi à partir d'un catalogue de processus. Ce dernier ne requiert pas d'être exhaustif car nous proposons une spécialisation dynamique (i.e. nous générons des variantes sur demande (*on the fly*)). De plus, nous proposons une approche de spécialisation générique. En effet, notre technique de spécialisation se base sur des questions et des transformations indépendantes de tout domaine d'affaires.

Ce chapitre est dédié à la présentation de la vue globale de notre méthodologie de réutilisation et d'adaptation de processus d'affaires. La section 3.1 présente les principes de base de notre approche. La section 3.2 décrit une première démarche que nous avons adoptée. Nous présentons par la suite les problèmes qui lui sont liés. La section 3.3 introduit notre

nouvelle approche de spécification de processus d'affaires générique à base d'une ontologie d'affaires. La section 3.4 illustre la démarche de validation de notre approche.

3.1 Une nouvelle approche de spécification générique

3.1.1 Principes

L'objectif principal de notre travail est de développer une nouvelle méthodologie et des outils pour la réutilisation et l'adaptation de processus génériques selon les besoins spécifiques des organisations. Nous visons une approche mixte qui combine le catalogage et la génération de processus d'affaires. Donc, (i) nous proposons un catalogue de processus génériques qui peut servir de point de départ. On peut organiser ce catalogue par une classification multi-facettes ou selon le manuel MIT (Malone *et al.*, 1999), et (ii) on génère des variantes sur demande (*on the fly*) en se basant sur les questions/réponses que l'on pose à l'analyste d'affaires. La clé de l'approche est de trouver une façon de généraliser les questions, jusque-là liées aux processus spécifiques dans les approches existantes, et de généraliser les transformations en question. Les principaux avantages de cette approche sont : (i) un catalogue non exhaustif de processus d'affaires, (ii) des opérateurs de spécialisation qui s'appliquent à des processus provenant de différents domaines d'affaires. Idéalement, le catalogue de processus d'affaires doit offrir une hiérarchie facile à naviguer et l'approche doit être facile à assimiler par des utilisateurs ayant seulement des connaissances de domaines d'affaires spécifiques. La Figure 3.1 montre le processus de l'approche globale que nous proposons pour la réutilisation et l'adaptation de processus d'affaires.

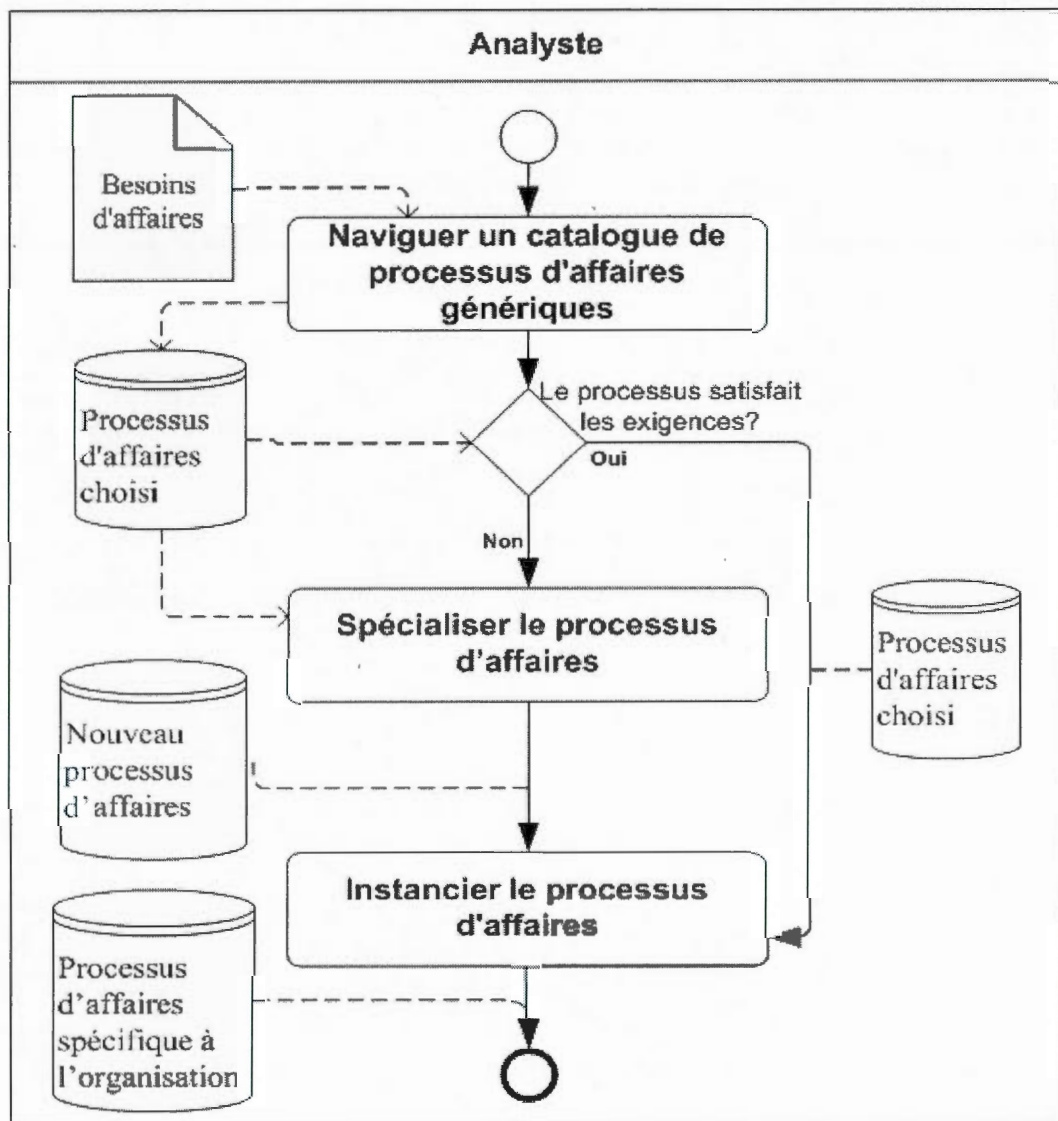


Figure 3.1 Les étapes de notre approche.

Pour passer d'un processus d'affaires générique à un processus spécifique, nous proposons une approche de spécialisation par les questions qui suit le processus suivant : à partir d'un catalogue de processus d'affaires génériques et des besoins d'affaires spécifiques à une organisation, l'analyste obtient, par navigation, un processus qui répond le mieux à ses besoins. Par la suite, l'analyste peut décider d'utiliser le processus obtenu tel quel ou le spécialiser. La spécialisation suivra une approche transformationnelle générique dirigée par

des questions. Finalement, ce processus sera instancié pour obtenir un processus spécifique à l'organisation.

3.1.2 Exemple

Dans la suite de ce chapitre, nous proposons d'étudier l'exemple d'un processus d'approvisionnement pour décrire notre approche. Un processus d'approvisionnement consiste à fournir à une entreprise donnée les ressources et les services nécessaires auprès de sources d'approvisionnement, et ceci en temps voulu. Le processus d'approvisionnement commence par remplir une réquisition d'achat qui est ensuite soumise à une approbation budgétaire. Une fois la demande approuvée, le service d'achat doit sélectionner un fournisseur pour le produit. Par la suite, il envoie une commande d'achat au fournisseur sélectionné. Les dernières étapes consistent en la réception du produit et le paiement du fournisseur. La Figure 3.2 montre le modèle de la vue fonctionnelle simplifiée du processus d'approvisionnement.

Cette vue fonctionnelle (fig. 3.2) est indépendante du domaine d'application (i.e. domaine d'affaires spécifique). À ce niveau d'abstraction, ce processus peut être utilisé pour commander des crayons, des ordinateurs ou des avions. Il y a beaucoup de variations de ce processus indépendamment du domaine d'affaires spécifique des organisations. Par exemple, pour des achats en ligne, nous devons généralement payer avant de recevoir le produit. Deuxièmement, si l'acheteur a conclu un contrat en cours de validité avec un fournisseur, alors il n'a pas à sélectionner un partenaire d'affaires à chaque fois. Troisièmement, si le demandeur est aussi le décideur, il n'a pas besoin de demander l'approbation.

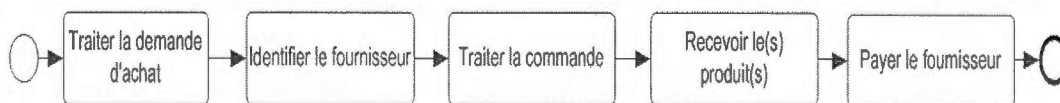


Figure 3.2 La vue fonctionnelle simplifiée du processus d'approvisionnement.

Toutes ces variations ne dépendent pas d'une industrie cible. De là, les deux questions que nous posons :

1. Y-a t-il un moyen d'organiser les processus d'affaires selon une hiérarchie de spécialisation qui facilite aux utilisateurs la navigation et la sélection du processus d'affaires qui convient le mieux à leur organisation ?
2. Y-a t-il un moyen de générer certaines de ces spécialisations dynamiquement sur la base de certains catalogues de spécialisations primaires ?

La Figure 3.3 montre le modèle simplifié de la vue informationnelle du processus d'approvisionnement. Les entités *Requester* et *Supplier* représentent l'acheteur et le fournisseur respectivement. L'entité *Order* décrit la commande d'achat. Les lignes de la commande sont représentées par *LineItem*. Comme on vient de le mentionner, le processus dépend de l'existence d'un contrat commercial entre l'acheteur et le fournisseur. Deuxièmement, la réception du produit dépend de la nature du produit. Par exemple, il peut s'agir d'un produit tangible (e.g. une chaise, un ordinateur, un livre, etc.) ou d'un produit non tangible (e.g. l'accès Internet, service téléphonique, etc.). La Figure 3.4 montre un nouveau modèle de la vue informationnelle qui accommode le deuxième point de variation (i.e. si le produit est tangible ou non).

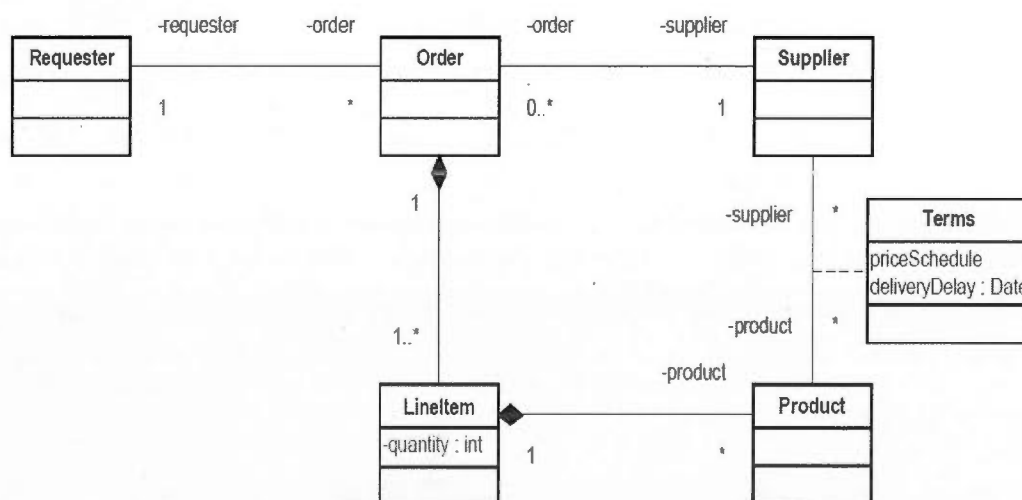


Figure 3.3 La vue informationnelle du processus d'approvisionnement.

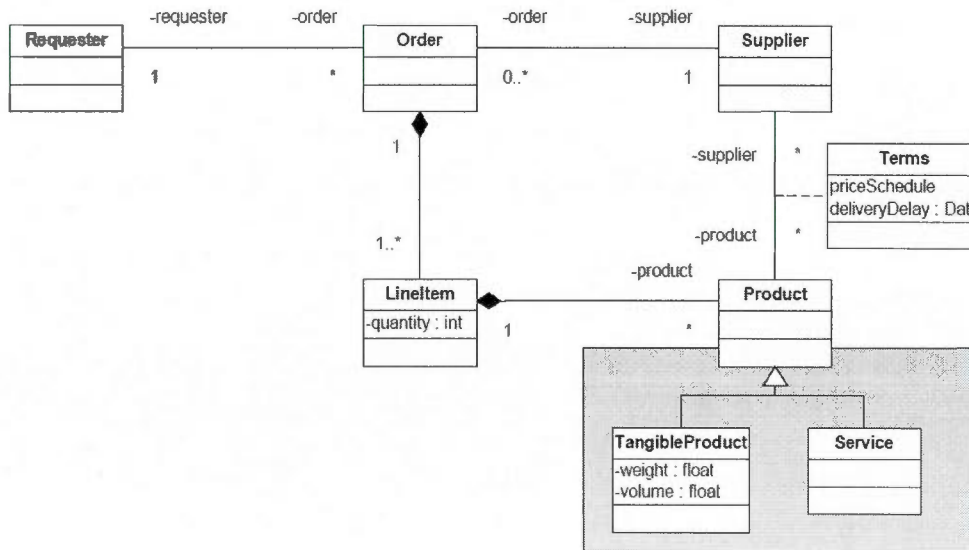


Figure 3.4 La vue informationnelle après la spécialisation de l'entité produit.

Dans de ce travail, nous présentons une nouvelle méthodologie de spécialisation automatique de processus d'affaires. Nous proposons une approche par question (sect. 2.4). Nous visons le développement de questions et de transformations génériques qui peuvent s'appliquer à des processus provenant de différents domaines d'affaires. Nous allons présenter une première démarche de réutilisation de processus d'affaires que nous avons adoptée au début de ce projet de recherche. Nous l'appelons *l'approche métalangage*. Nous présentons ses problèmes inhérents dans la section 3.2.4. Par la suite, nous décrivons notre nouvelle approche dite *ontologique*.

3.2 L'approche métalangage pour la spécification de processus d'affaires

3.2.1 Principes

En parcourant les processus du manuel MIT ainsi que les questions de spécialisation utilisées, nous avons réalisé que les questions sont reliées aux différents rôles et acteurs impliqués dans

le processus (e.g. client, fournisseur, etc.), à la nature des ressources produites et consommées par les activités (e.g. service, tangible, etc.) et à l'organisation où les activités se déroulent. Dès lors, dans un premier travail, nous avons proposé la démarche de généralisation suivante : (i) généraliser les questions spécifiques et les relier à des éléments du méta-modèle (fig. 3.5), et (ii) généraliser les opérateurs de transformations pour qu'ils s'appliquent aux éléments du méta-modèle. La construction des questions génériques se fait à partir de questions spécifiques en remplaçant chaque élément de modèle par la classe du méta-modèle correspondante (i.e. méta-type). Lors de la spécialisation, les questions ainsi générées seront instanciées pour des processus spécifiques, et les opérations de transformations génériques seront appliquées aux éléments correspondants du modèle de processus à spécialiser. Par la suite, il faut prévoir l'effet d'une réponse (i.e. effet d'une réponse positive ou négative dans le cas d'une question binaire) sur les différentes vues d'un processus de manière générique. Ces réponses spécifiques vont permettre de spécialiser le processus générique par le biais d'opérateurs de spécialisation génériques (Mili *et al.*, 2009).

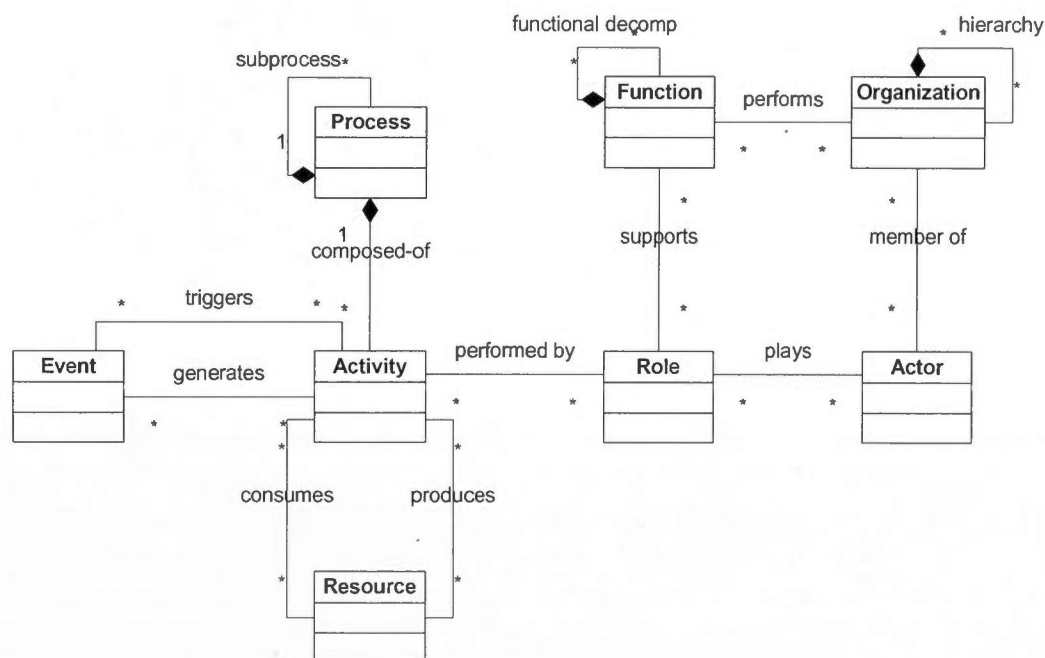


Figure 3.5 Un méta-modèle primaire de processus d'affaires. (Source : Mili *et al.*, 2009)

Pour illustrer le concept de généralisation et d'instanciation de l'approche métalangage, prenons l'exemple de la question spécifique : *est-ce qu'il y a un contrat entre l'acheteur et le fournisseur ?* La transformation spécifique dans le cas d'une réponse positive consiste à ajouter une entité contrat entre l'acheteur et le fournisseur au niveau informationnel, et enlever les étapes de recherche du fournisseur aux niveaux fonctionnel et dynamique. La généralisation de cette question donne la question générique suivante : *est-ce qu'il y a un contrat entre <Acteur> et <Acteur> ?* La transformation générique correspondante dans le cas d'une réponse positive consiste à ajouter un contrat entre <acteur1> et <acteur2> au niveau informationnel, et enlever les étapes de recherche du partenaire d'affaires aux niveaux fonctionnelle et dynamique.

Dans cette approche, les questions de spécialisation sont formulées sur la base des entités et des associations d'un méta-modèle de processus d'affaires. Ces questions génériques vont être applicables à des processus génériques. Pour chaque processus, il faut instancier ces questions dépendamment du domaine du processus pour obtenir des questions spécifiques.

3.2.2 Qu'est-ce qu'un processus d'affaires générique

Un processus d'affaires générique, ou encore processus d'affaires commun (*Common Business Process*) est un processus d'affaires commun à un domaine d'affaires. Ce type de processus est indépendant du secteur d'industrie spécifique (e.g. vente au détail, industrie pharmaceutique, etc.). Cet aspect de généralité permet la réutilisation de processus dans divers contextes pour différentes organisations. Notre hiérarchie de spécialisation se base sur un processus générique pour chaque domaine d'affaires. Ce processus servira comme référence à toutes ses variantes. Nous appelons *la famille d'un processus générique* l'ensemble de toutes ses variantes. Dans le chapitre 5, nous présentons les stratégies les plus connues pour la conception d'un processus générique, à savoir, la stratégie (i) du processus maximal, (ii) du processus minimal, (iii) du processus standard, (iv) du processus le plus utilisé, et finalement, (v) du processus ayant la distance moyenne minimale parmi les variantes.

3.2.3 Représentation de processus d'affaires

La représentation de processus d'affaires doit ressortir les quatre vues identifiées par Curtis, Kellner et Over (1992), soit, les vues fonctionnelle, dynamique, informationnelle, et organisationnelle. La Figure 3.6 montre le modèle des vues de processus d'affaires. Chaque vue (*ProcessModelView*) consiste en un ensemble d'éléments du modèle (*ModelElement*) qui lui sont propres : (i) la vue fonctionnelle (*FunctionalView*) représente les activités à accomplir et les ressources, (ii) la vue dynamique (*DynamicView*) met l'accent sur le concept d'orchestration, (iii) la vue informationnelle (*InformationalView*) décrit la structure des entités manipulées par un processus d'affaires, et enfin, (iv) la vue organisationnelle (*OrganizationalView*) représente où et par qui les activités sont réalisées, la structure organisationnelle, les rôles et les mécanismes de communication. L'entité *ViewSpecialization* représente la trace de spécialisation d'un type de vue après une réponse à une question.

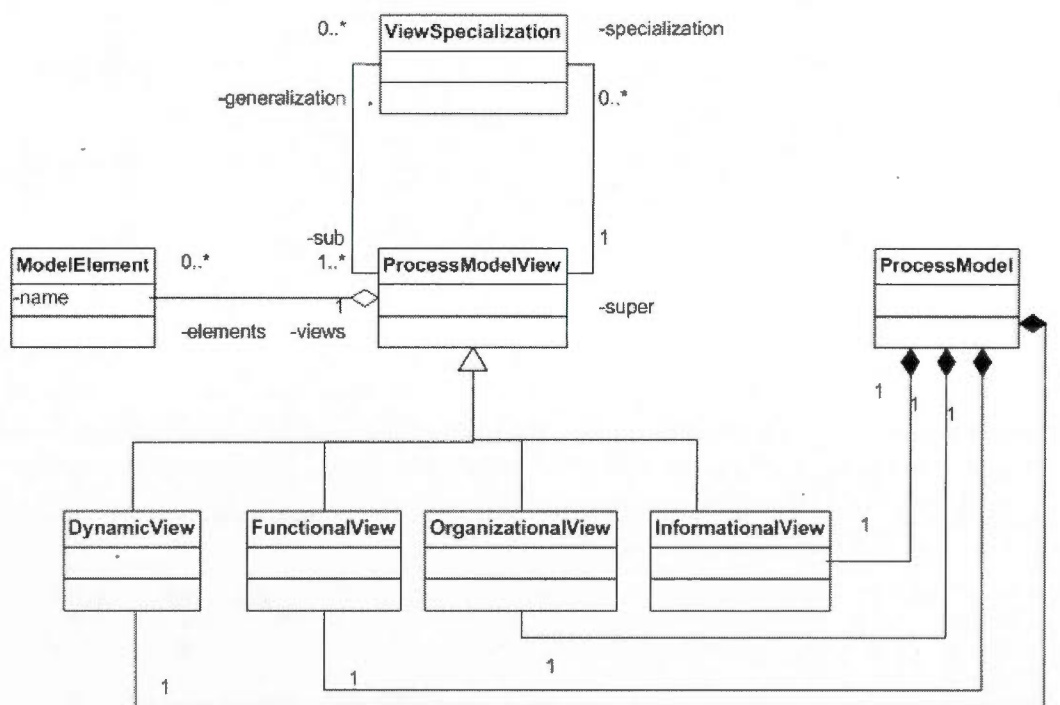


Figure 3.6 Le modèle des vues d'un processus d'affaires.

3.2.4 Représentation des questions

Nous considérons une question générique comme une fonction qui prend des paramètres et retourne une valeur. Les types des paramètres sont des entités du méta-modèle. La valeur de retour de la fonction est la réponse à la question. Prenons l'exemple de la question générique suivante : *Est-ce que le déroulement du processus est régi par un contrat entre <Acteur> et <Acteur> ?* On peut considérer cette question comme étant la fonction suivante :

Boolean processExecutionFollowsAContract(Actor actor1, Actor actor2) où <Actor> est un élément du méta-modèle.

Cette question peut être instanciée pour différents processus d'affaires afin d'obtenir des questions spécifiques. Dans notre exemple de processus d'approvisionnement (fig. 3.3), nous retrouvons deux acteurs : « **Requester** » et « **Supplier** ». L'ensemble des questions spécifiques contient toutes les combinaisons possibles d'association de ces deux paramètres. De là, on obtient les quatre questions spécifiques :

1. Est-ce que le déroulement du processus est régi par un contrat entre « Requester » et « Requester » ?
2. Est-ce que le déroulement du processus est régi par un contrat entre « Requester » et « Supplier » ?
3. Est-ce que le déroulement du processus est régi par un contrat entre « Supplier » et « Requester » ?
4. Est-ce que le déroulement du processus est régi par un contrat entre « Supplier » et « Supplier » ?

Plusieurs questions spécifiques sont clairement non pertinentes. Par exemple, les questions (1) et (4) semblent non pertinentes. De plus, cette question générique est symétrique dans le sens où $f(x, y) = f(y, x)$ pour deux paramètres x, y donnés. Les questions spécifiques (2) et (3) sont donc équivalentes. À partir de là, nous retiendrons une seule question spécifique : *Est-ce que le déroulement du processus est régi par un contrat entre « Requester » et « Supplier » ?*

Notre représentation des questions génériques comprend un filtre qui permet d'éliminer les questions inacceptables et redondantes. La Figure 3.7 montre notre modèle de représentation des questions génériques et spécifiques.

La classe *GenericQuestion* contient les attributs *name*, *description*, *core* et *filter*. L'attribut *filter* est un filtre qui permet d'éliminer les questions inacceptables et redondantes. L'attribut *core* représente le texte de la question générique en langage naturel sous forme de gabarit. Pour notre exemple, le texte est : *Est-ce que le déroulement du processus est régi par un contrat entre {0} et {1} ?* Où {0} et {1} sont des paramètres positionnels qui font référence aux éléments du méta-modèle. Ces paramètres sont représentés par la classe d'association *Parameter* dont le type est défini par la classe *ModelElementType* qui est un élément du méta-modèle. La classe *Question* représente la question spécifique instanciée à partir de *GenericQuestion* par une mise en correspondance (classe *Binding*) des paramètres de la question générique aux éléments du modèle du processus. Chaque réponse est représentée par la classe *Answer* qui peut spécialiser éventuellement une ou plusieurs vues d'un processus d'affaires.

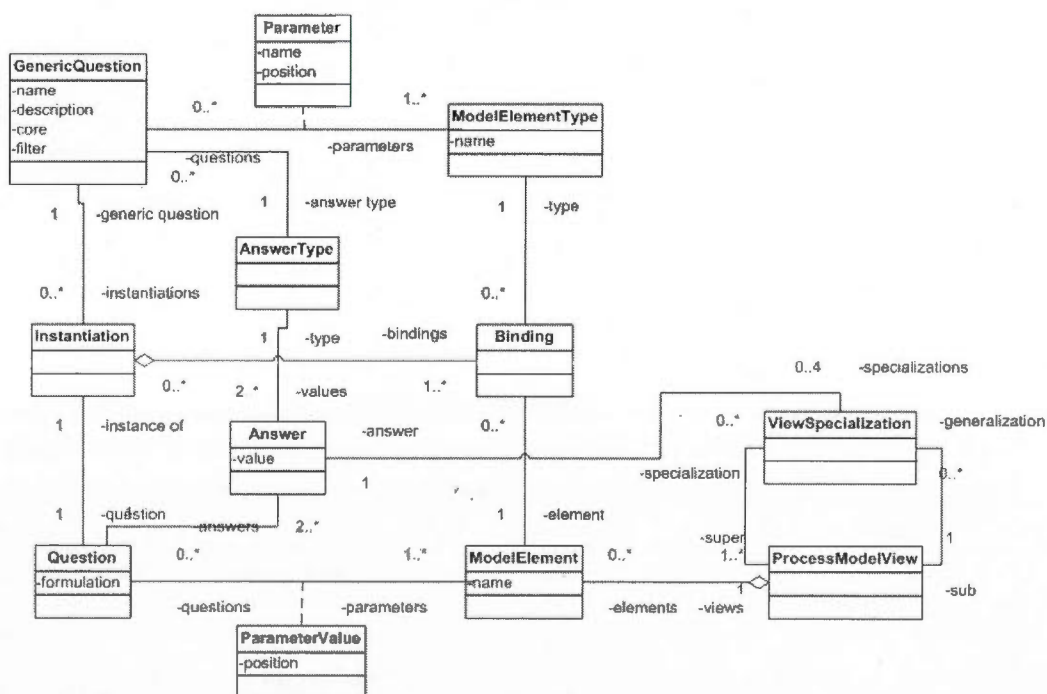


Figure 3.7 Le modèle de représentation des questions.

3.2.5 Représentation des spécialisations

Nous avons montré dans la section précédente qu'une réponse à une question spécifique peut spécialiser zéro, une ou plusieurs vues d'un processus d'affaires (voir fig. 3.7). Nous considérons la spécialisation comme l'application d'une série de transformations génériques aux modèles des vues d'un processus d'affaires. Ces transformations sont génériques puisqu'elles ne dépendent pas d'un domaine d'affaires particulier. Par contre, elles sont spécifiques aux types des vues, puisque chaque type de vue (fonctionnelle, dynamique, informationnelle et organisationnelle) a ses propres règles de transformations. Par exemple, une réponse positive (i.e. Oui) à notre question de contrat provoquera les transformations suivantes à la vue informationnelle :

- 1 Créer une nouvelle classe `Contract`;
- 2 Créer une nouvelle association bidirectionnelle entre les classes `Contract` et `Requester` en spécifiant les cardinalités et les rôles;
- 3 Créer une nouvelle association bidirectionnelle entre les classes `Contract` et `Supplier` en spécifiant les cardinalités et les rôles.

La Figure 3.8 montre notre modèle de spécialisation de processus d'affaires. La classe `TransformationRule` représente une règle de transformation. Tel qu'illustré, une règle de transformation est liée à une question générique, une réponse spécifique, et un type de vue de processus. Ce sont ces transformations-là qui constituent le défi le plus important et le plus difficile de notre approche. L'entité `ViewSpecialization` représente la trace d'exécution des règles de transformation appliquées à un type de vue. La relation `trace` est représentée par l'association plusieurs à plusieurs entre `TransformationRule` et `ViewSpecialization`.

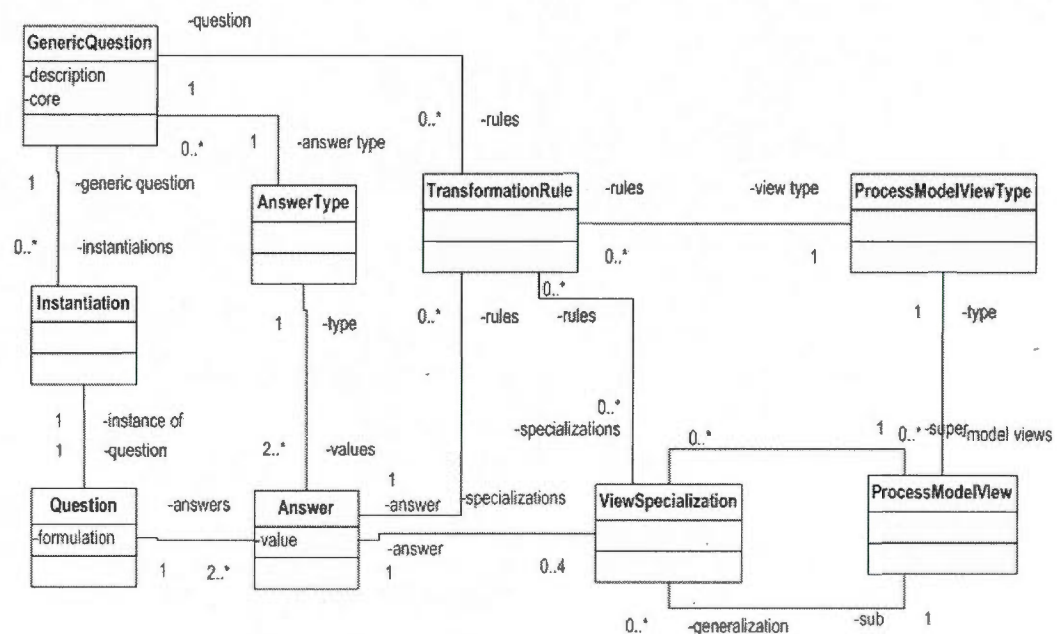


Figure 3.8 Le modèle de spécialisation des processus d'affaires.

3.2.6 Implémentation

Dans cette section, nous décrivons brièvement l'implémentation des différents concepts de l'approche métalangage. Nous commencerons d'abord par notre méta-modèle de processus d'affaires. Par la suite, nous décrivons l'implémentation des questions génériques. Finalement, nous présentons la mise en œuvre de la spécialisation avec un système de gestion de règles.

3.2.6.1 Implémentation du méta-modèle de processus d'affaires

Nous avons implémenté notre méta-modèle de processus d'affaires avec EMF™ (*Eclipse Modeling Framework*) (Eclipse, 2008). EMF est un cadre de modélisation qui implémente EMOF (*Essential Meta Object Facility*). Il est destiné à l'environnement de développement Eclipse™. Nous présentons le cadre EMF dans le chapitre 5. La Figure 3.9 montre une partie de notre méta-modèle mis en œuvre avec le cadre EMF.

3.2.6.2 Implémentation des questions

Nous avons implémenté notre méta-modèle de questions génériques avec un schéma XML. La Figure 3.10 montre la représentation de la question du contrat : *Est-ce que le déroulement du processus est régi par un contrat entre <RolePerformer> et <RolePerformer> ?* où <RolePerformer> est un élément de notre méta-modèle.

La balise <core> définit le texte de la question. {0} et {1} sont les paramètres positionnels de la question. Ces paramètres se trouvent entre les balises <arguments> où l'attribut *filter* représente le filtre d'instanciation. Le concept de filtrage permet d'éviter la prolifération des questions spécifiques. Pour la question du contrat, le filtre consiste à spécifier une liste de propriétés mathématiques qui permettent de réduire le nombre de questions spécifiques. La conception de filtres génériques qui permettent de réduire le nombre de n-uplets d'instanciation de nos questions génériques représente un autre défi important de cette approche. En effet, nous envisageons une sémantique de filtrage plus complexe qui devrait nécessiter une représentation plus élaborée.

```
<question>
  <name>Binding agreement between role performers</name>
  <description>
    The process follows an agreement between role performers
  </description>
  <core answerType="java.lang.Boolean" >
    Does the process execution follow an agreement between {0} and {1}?
  </core>
  <arguments filter ="!reflexive, symmetric">
    <argument>RolePerformer</argument >
    <argument>RolePerformer</argument >
  </arguments >
</question>
```

Figure 3.10 Exemple de représentation d'une question générique.

3.2.6.3 Implémentation de la spécialisation

Nous avons implémenté la spécialisation avec Drools (JBoss, 2012). Drools est un système de gestion de règles d'affaires conçu pour le langage Java. Une règle Drools se compose d'une partie condition après le mot clé *when* et une partie action après le mot clé *then*. Nous présentons Drools dans le chapitre 6 qui sera dédié à la spécialisation de processus d'affaires. La Figure 3.11 montre la règle de spécialisation de la vue informationnelle dans le cas d'une réponse favorable à la question du contrat : *Est-ce que le déroulement du processus et régi par un contrat entre <Requester> et <Supplier> ?*

La Figure 3.12 montre le modèle informationnel avec EMF qui résulte de l'application de la règle de transformation de la Figure 3.11 au modèle de la vue informationnelle du processus d'approvisionnement (fig. 3.3).

```
rule "AGREEMENT IN THE PROCESS-INFORMATIONAL VIEW"
when
    $q : Question(answer.value=="true",
                  genericQuestion.name==Questions.AGREEMENT_IN_THE_PROCESS);

then
    // Add a new Class , dateFrom and dateTo attributtes
    clazz=inf_view.addEClass("Contract",BpPackage.eINSTANCE.getResourceProduct());
    inf_view.addEAttribute(clazz,"dateFrom", ecorePackage.getEDate());
    inf_view.addEAttribute(clazz,"dateTo", ecorePackage.getEDate());
    // Add bidirectional reference to the first entity (parameter 1)
    ref=inf_view.addEReference(clazz, $q.param(0), $q.param(0).getName(),1, 1,null);
    inf_view.addEReference($q.param(0),clazz, "contract",0, ETypedElement.UNBOUNDED_MULTIPPLICITY,ref);
    // Add bidirectional reference to the second entity (parameter 2)
    ref=inf_view.addEReference(clazz,$q.param(1), $q.param(1).getName(),1, 1,null);
    inf_view.addEReference($q.param(1),clazz, "contract",0, ETypedElement.UNBOUNDED_MULTIPPLICITY,ref);
    // Informational view name, nsPrefix and nsURI are auto updated when persisting
    inf_view.persist();
end
```

Figure 3.11 Exemple de règle de transformation.

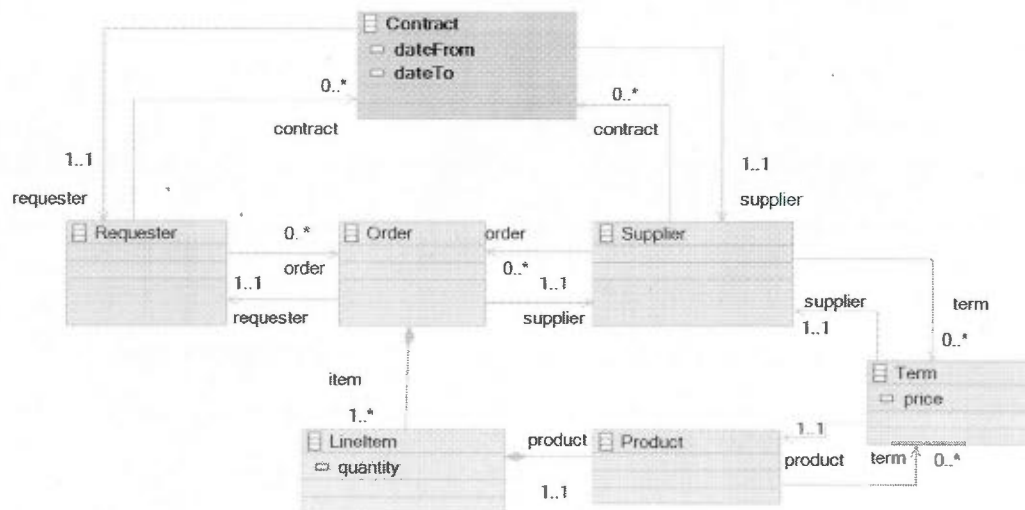


Figure 3.12 La vue informationnelle (modèle Ecore) après la transformation.

3.2.7 Validation de l'approche métalangage

Nous avons montré dans cette section que notre approche métalangage est réalisable. En effet, au sein de ce travail, nous avons présenté une nouvelle démarche pour la spécification de processus d'affaires. Nous avons développé des modèles de questions et de transformations des différentes vues de processus d'affaires. Dans ce même travail, nous avons appuyé notre approche de transformation par un prototype de mise en œuvre plausible des concepts présentés, notamment, la mise en œuvre de la représentation des processus d'affaires, de l'instanciation des questions et des règles de spécialisations. Finalement, nous avons pu valider notre méthodologie en spécialisant des processus d'affaires de logistiques, à savoir, le processus d'achats, de vente, de distribution et de production. Ces spécialisations utilisaient des questions parmi notre ensemble des quinze questions génériques. Toutefois, afin de valider notre approche, nous devons être en mesure de l'appliquer pour la majorité de nos quinze questions génériques, et cela, sur un ensemble plus élargi de processus d'affaires.

Après avoir analysé un ensemble restreint de processus seulement, il est apparu que:

1. Les questions qui avaient un sens pour un groupe de processus n'ont guère de sens pour d'autres. Par exemple, la question sur la nature de la ressource (i.e. tangible ou non) a un sens pour les processus de logistique mais n'a pas de sens pour les

processus de ressources humaines et de finances.

2. Pour ces questions qui s'appliquent à des groupes de processus, les spécialisations dépendent souvent de ces groupes de processus.
3. En plus des difficultés énumérées ci-dessus, la vue dynamique des processus d'affaires est souvent difficile à spécialiser.
4. Lors de la généralisation des questions sur la base des entités et des associations du méta-modèle de processus d'affaires, celles-ci perdent leur sémantique.

Prenons ce dernier point par exemple. Notre validation a montré que lorsque les questions perdent leur sémantique, il en résulte deux inconvénients majeurs : (1) la méthode génère beaucoup de questions superflues, et (2) il est très difficile de trouver des spécialisations génériques qui ont du sens pour différents groupes de processus d'affaires.

Pour le premier point, prenons l'exemple de la question du contrat : *Est-ce que le déroulement du processus est régi par un contrat entre « Acteur » et « Acteur » ?* Nous devons remplacer tous les éléments du méta-modèle par des éléments du modèle du domaine en entrée dont la généralisation est l'élément du méta-modèle. Dans notre exemple de processus d'approvisionnement, nous retrouvons deux acteurs, soit, *l'acheteur* et *le fournisseur*. L'ensemble de questions spécifiques contient toutes les combinaisons possibles d'association des ces éléments. Ce qui donne 2^2 combinaisons possibles.

Nous pouvons généraliser la règle pour chaque paramètre de la question générique. Par la suite, nous généralisons pour tous les paramètres. Soit Q une question générique; X la cardinalité de l'ensemble correspondant à un paramètre P de Q . Soit Y la cardinalité de l'ensemble des éléments du modèle du processus d'affaires cible (i.e. celui qu'on veut spécialiser) dont la généralisation est P . On obtient Y^X combinaisons de questions spécifiques pour ce paramètre. Il faut appliquer la même technique pour chaque type de paramètre de la question. Le nombre de questions spécifiques sera obtenu par le produit de toutes les questions obtenues pour chaque paramètre.

Soit x_1, x_2, \dots, x_n la cardinalité de l'ensemble de paramètres pour une question générique. Soit y_1 et y_2, \dots, y_n la cardinalité de l'ensemble des éléments du modèle cible

correspondant au paramètre $x_1, x_2 \dots x_n$ respectivement. Alors, le nombre de questions spécifiques N est obtenu par la formule : $N = \prod_{i=1..n} Y_i^{x_i}$

Cette version de l'approche exige la conception d'une technique de filtrage complexe afin d'éviter la prolifération des questions spécifiques. En effet, les questions génériques requièrent une bonne compréhension du sens métier. Par exemple, pour la question du contrat, l'ordre de remplacement n'est pas important du fait que le contrat entre « acteur1 » et « acteur2 » est équivalent à contrat entre « acteur2 » et « acteur1 ». Nous parlons de symétrie dans ce cas. Nous pouvons aussi remarquer que la répétition n'est pas acceptée du fait que la question d'un contrat entre le même acteur n'a pas de sens.

Le deuxième problème réside dans le fait qu'il est très difficile de trouver des spécialisations génériques applicables et qui ont du sens pour des processus spécifiques. Prenons l'exemple d'une des questions de cette première approche : *Est-ce que le <Rôle> et le <Rôle> peuvent être joués par le même <Acteur> ?* Un acteur peut effectuer plusieurs rôles dans une organisation. Cela influence le déroulement du processus étant donné que les activités effectuées avec ces rôles peuvent se combiner ou être simplifiées. Il est difficile de définir des règles de spécialisation génériques à partir de cette question. De plus, la question manque de sens métier lorsqu'elle est appliquée pour un processus spécifique.

Notre approche métalangage exprime les questions en termes des entités et des associations du méta-modèle de processus d'affaires. Ces questions sont formulées en utilisant la théorie des grammaires de cas (*case grammar*) de Charles Fillmore (1968). Cette approche expose des questions trop génériques. De plus, il est très difficile de trouver des opérateurs de spécialisation génériques et réutilisables.

Nous ne pensons pas que ces difficultés invalident l'idée de spécialisation par les questions. Toutefois, ces difficultés présentent un défi de taille pour déterminer un ensemble fini de questions génériques et d'opérateurs de spécialisation génériques qui permettrait de gérer une variété de processus susceptibles d'être rencontrés dans la pratique. Dans la prochaine section, nous montrons pourquoi notre première approche n'a pas donné les résultats escomptés. Nous proposons alors une deuxième approche plus prometteuse que nous avons adoptée par la suite.

3.2.8 Le problème de base de l'approche métalangage

Jusqu'ici, nous avons été informels sur le terme de « généralisation » des questions. Cette généralisation nous a permis d'aller des questions spécifiques vers des questions génériques. Pour être plus précis, nous avons utilisé une opération de classification le long du langage. En effet, la relation entre les questions spécifiques et les questions génériques est celle entre une instance et une classe, et non pas celle entre une classe et une superclasse. De même, lorsque nous appliquons une question générique à un processus, l'opération utilisée est une instanciation, par opposition à une spécialisation. En effet, de la question spécifique vers la question générique, nous remplaçons l'élément du modèle de processus d'affaires, par exemple le « fournisseur » dans un processus d'approvisionnement, par la classe du méta-modèle à laquelle il appartient, dans ce cas, < **Acteur** >. L'opération inverse est effectuée lors de l'application d'une question générique à un processus d'affaires donné. Dans ce cas, nous remplaçons les éléments du méta-modèle (i.e. méta-classes) par des instances spécifiques au processus.

Pour illustrer la différence entre la généralisation et la classification, supposons que nous voulions modéliser le processus d'affaires d'un fabricant d'électronique. Un processus de production doit manipuler un appareil électronique, représenté par sa tension, puissance, classe FCC (*Federal Communications Commission*), etc. Si nous modélisons une entreprise agricole, un fruit ou un légume serait caractérisé par sa couleur, son poids, sa taille, sa saveur, teneur en sucre, et ainsi de suite. Dans ce cas, il n'y pas de généralisation utile entre un dispositif électronique et un fruit. Ils n'ont pas d'attributs en commun. Toutefois, si on les modélise comme des instances d'un type de produit, qui se caractérise par un coût de production, des exigences d'entreposage, d'une durée de vie et un prix de vente prévu, nous trouverons des similarités communes et utiles entre les processus qui les manipulent. En fait, c'est ainsi que nous sommes en mesure de définir les processus de logistique qui ne dépendent pas du produit manipulé.

Cette analyse nous mène au travail de Mili et Pachet (2000) qui montre comment classer un concept selon plusieurs dimensions. Par exemple, la classe *CheckingAccount* de la Figure 3.13 peut être classifiée selon deux dimensions :

1. Dans la Figure 3.14-b, nous montrons le modèle de classe de la Figure 3.13 comme une phrase du langage de modélisation, dans notre cas, UML. Ainsi, les classes `CheckingAccount` et `Customer` sont des classes. De même, l'association `holds` entre `CheckingAccount` et `Customer` est représentée par deux instances de `AssociationEnd`. Cette classification est dite: **Aller méta le long de la dimension du langage** (*going meta along the language dimension*).
2. Dans la Figure 3.14-a, la classe `CheckingAccount` est vue comme un `BankingProductType`. Chaque type de produit bancaire peut être défini par un ensemble de restrictions (e.g. si un solde négatif est autorisé), des privilèges (e.g. des transactions gratuites, rabais, etc.), et une grille tarifaire (e.g. par retrait, par chèque, etc.), et être disponible seulement à certains types de clients. Ce niveau de représentation est appelé **niveau de nomenclature** (*nomenclature level*), où les produits et services sont définis par opposition au modèle de niveau instance qui est utilisé pour modéliser les instances de ces produits et services pour les clients réels. Ce méta-modèle s'applique à une variété de domaines d'affaires et expose une sémantique d'affaires intéressante, à savoir, la définition de types de produits, les types de clientèle, et la relation entre eux.



Figure 3.13 Modèle UML simple

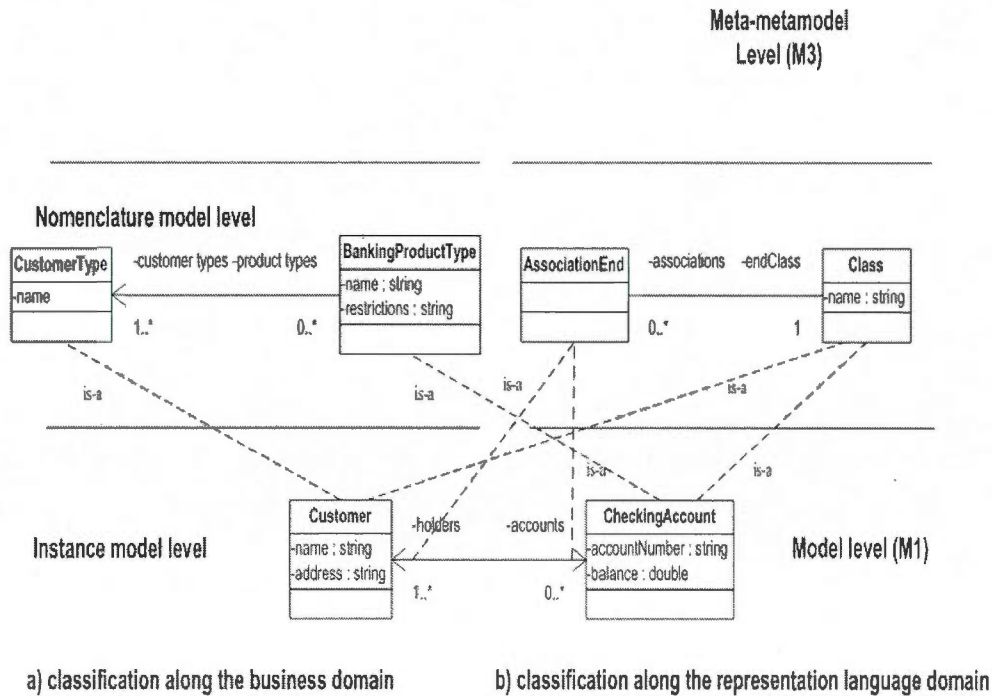


Figure 3.14 Les niveaux de classification des concepts.

3.2.9 Vers une approche à base d'une ontologie métier

Dans notre première approche, aller méta de la question spécifique vers la question générique n'est pas nécessairement faux, mais aller méta vers le méta-modèle du processus d'affaires au niveau de la dimension sémantique d'affaires n'est pas correct. Concrètement, au lieu de se concentrer sur le fournisseur et l'acheteur comme étant des acteurs dans un processus d'approvisionnement, nous devrions essayer de trouver à quel type de processus d'affaires appartient le processus d'approvisionnement. Et de là, essayer de trouver des points communs avec d'autres processus appartenant à différents groupes de processus d'affaires. Par exemple, il est plus intéressant de voir les processus d'approvisionnement, de vente, de recrutement et de finance, comme des processus où des ressources sont échangées dans un contexte économique plutôt que dire que le fournisseur, l'acheteur, l'employeur et l'employé sont tous des acteurs qui jouent différents rôles dans un processus d'affaires. Par conséquent, nous cherchons une classification à base d'une ontologie de domaine (processus d'affaires) riche qui nous permettrait de voir les similitudes entre des processus de différents domaines

d'affaires. Une solution serait d'utiliser une ontologie d'affaires. Ce type d'ontologie perçoit un processus d'affaires en termes de création et d'échange de valeurs économiques. Par exemple, quand on généralise la proposition *"acheteur achète le produit d'un vendeur"*, nous obtenons dans un niveau méta-métier à base d'une ontologie d'affaires, *"Agent1 échange une ressource contre de l'argent avec Agent2"*, plutôt que *"Rôle A effectue activité qui utilise une ressource avec rôle B"* au niveau métalangage. La proposition méta au niveau ontologique a plus de sens que celle du niveau métalangage de notre première approche. À partir de cette construction, on peut étudier cet échange et examiner, de façon générique, si l'échange de ressources implique par exemple un contrat entre les agents. Ce qui impliquerait des changements connus au niveau des modèles de processus d'affaires.

La section suivante décrit notre nouvelle approche ontologique pour la spécification de processus d'affaires. Nous décrivons les principes de cette nouvelle approche. Par la suite, nous présentons l'ontologie d'affaires REA. Finalement, nous illustrons cette approche ontologique par un exemple.

3.3 L'approche ontologique de spécification de processus d'affaires

3.3.1 Principes

L'approche métalangage exprime les questions en termes d'un méta-modèle de processus d'affaires traditionnel. On a vu qu'en généralisant par un méta-modèle traditionnel, on perd la sémantique d'affaires. Pour remédier aux inconvénients de cette approche, il nous faut généraliser sur la dimension affaires en trouvant, par exemple, le point commun entre acheter un produit ou service, d'un côté, et engager une ressource (humaine) de l'autre. Ainsi, une question (et les spécialisations correspondantes) pourront être appliquées aux deux. Pour cela, il nous faut une ontologie d'affaires qui puisse faire abstraction des intervenants dans un échange, par exemple, et du produit échangé, et qui puisse faire ressortir ce que les processus ont de commun. Ainsi, autant les questions que les opérateurs de spécialisation auront un sens métier. En effet, une ontologie d'affaires est une ontologie qui souligne principalement le concept de création et d'échange de valeurs économiques (e.g. les ressources). Ce type

d'ontologie ajoute un niveau sémantique qui permet une meilleure compréhension de processus d'affaires (Hruby, 2006).

3.3.2 Le choix d'une ontologie d'affaires

Nous avons le choix entre trois ontologies d'affaires, REA (Geerts et McCarthy, 2000), l'ontologie e³-value (Gordijn et Akkermans, 2001) et e-BMO (*e-Business Model Ontology*) (Osterwalder et Pigneur, 2002). En l'absence d'une ontologie commune et acceptée, nous avons choisi l'ontologie REA puisqu'elle est simple et qu'elle couvre le concept d'échange et de transformation de ressources économiques. L'ontologie REA est devenue le fondement pour plusieurs normes d'échange électronique. De plus, certains (e.g. Hruby, 2006) ont identifié des patrons REA pour modéliser les "interactions d'affaires les plus communes". Ces patrons peuvent devenir l'objet de nos questions génériques et de nos opérateurs de spécialisation.

L'ontologie REA a été proposée par William E. McCarthy pour modéliser des systèmes d'information dans le domaine de la comptabilité (McCarthy, 1982). Elle est devenue, par la suite, une ontologie d'affaires (Geerts et McCarthy, 2000). Les composantes principales de l'ontologie REA sont : (1) les ressources économiques, (2) les événements économiques et (3) les agents économiques. Les ressources économiques sont des objets rares et utiles qui sont sous le contrôle de l'entreprise (McCarthy, 1982). Les événements économiques sont définis comme étant des phénomènes qui reflètent le changement au niveau des ressources suite à un échange, une création ou une transformation. Le méta-modèle de base de l'ontologie REA est présenté à la Figure 2.7 (voir sect. 2.2.5).

Les ressources sont incrémentées et décrémentées dans le contexte d'un événement économique. La relation de dualité représente le lien entre les événements économiques duals. Ce lien montre quelle ressource a été augmentée et quelle ressource a été diminuée en contrepartie. Le modèle différencie également les agents internes des agents externes dans le contexte d'un événement économique. Des concepts plus avancés sur l'ontologie REA seront présentés dans le chapitre 4.

3.3.3 Démarche de spécification de processus d'affaires

Pour spécifier un processus d'affaires générique à partir des besoins spécifiques des organisations, nous proposons le processus de spécialisation suivant :

- (i) À partir d'un processus d'affaires générique décrit avec la notation BPMN par exemple,
- (ii) on le décompose sous forme d'une chaîne de processus d'échange ou de conversion REA (voir sect. 2.2.5 et sect. 4.2). Cet ensemble de processus élémentaires interconnectés représente la chaîne de valeurs,
- (iii) on modélise chacun de ces processus selon les patrons de processus REA correspondants,
- (iv) on pose les questions sur chacun de ces processus REA,
- (v) on applique les opérateurs de spécialisation à chaque processus selon les réponses aux questions, et
- (vi) on revient vers le processus BPMN.

Dans le chapitre 6, nous illustrons notre processus de spécialisation avec plus de détails. Nous verrons, par exemple, que les modèles de processus d'affaires du référentiel sont sauvegardés sous forme déjà décomposée.

3.3.4 Exemple : Spécification du processus d'approvisionnement

Afin d'expliquer notre approche ontologique, nous allons reprendre l'exemple du processus d'approvisionnement. La Figure 3.15 montre le processus générique en BPMN avec plus de détails. Comme pour l'approche métalangage, nous présentons la spécialisation de ce processus après une réponse positive à la question d'existence d'un contrat entre les agents économiques « Requester » et « Supplier ».

Nous allons d'abord étudier le patron d'affaires d'un contrat commercial avec REA. Nous verrons par la suite que notre approche ontologique de spécification de processus d'affaires est basée sur des patrons d'affaires.

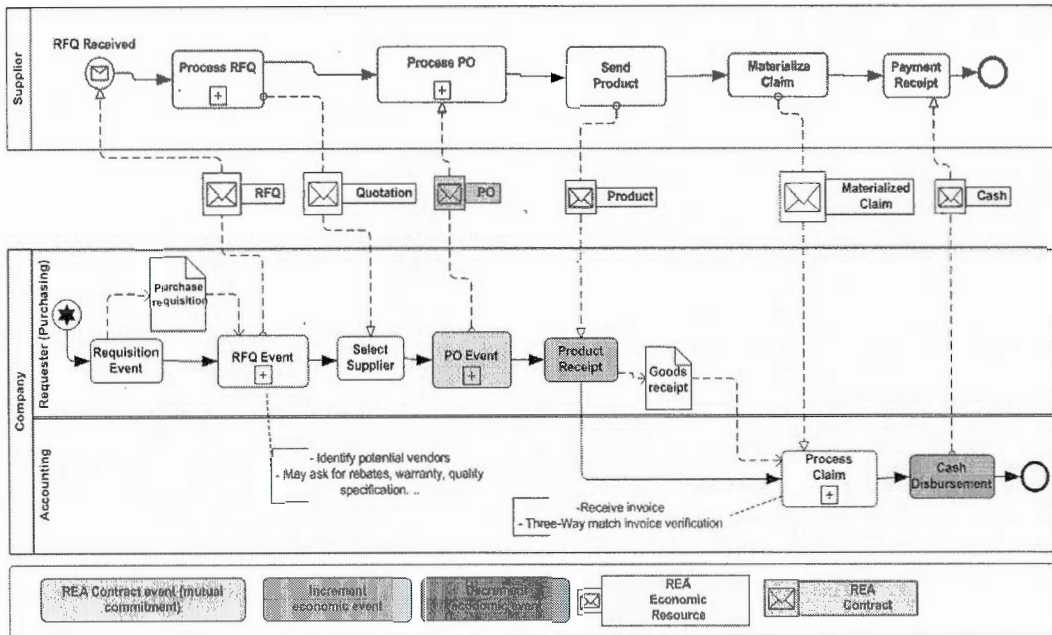


Figure 3.15 Processus d'approvisionnement.

3.3.4.1 La question du contrat REA

Un contrat est un accord entre deux parties ou davantage. Le contrat a pour objet l'établissement d'obligations à la charge ou au bénéfice de chacune de ces parties. Dans la terminologie REA, les obligations sont appelées engagements et les parties, des agents économiques. Le contrat et les engagements sont devenus des entités de l'ontologie REA.

Un contrat commercial impliquant plusieurs échanges entre les parties est un accord (*Agreement*) dans la terminologie REA. Un contrat REA a une durée de vie égale à celle du processus d'affaires. La Figure 3.16 illustre une version simplifiée du patron contrat.

En analysant ce patron d'affaires, nous pouvons examiner si l'échange de ressources économiques implique un accord entre les agents économiques. En effet, en nous basant sur le concept de cardinalité, nous constatons que l'accord entre les partenaires d'affaires n'est pas obligatoire. Son existence dépend du processus d'affaires des organisations. Dès lors, nous envisageons la question suivante : *Est-ce que le déroulement du processus {P} est régi par un accord ?*

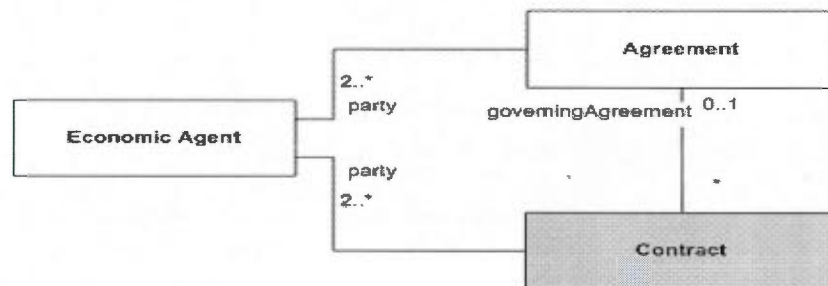


Figure 3.16 Une version simplifiée du patron contrat. (Source : Hruby, 2006)

Nous avons utilisé le cadre ontologique de REA pour identifier l'ensemble de nos questions génériques. Plus précisément, nos questions génériques se basent sur les patrons d'affaires REA de base et ceux développés par Pavel Hruby (2006). Dans la suite de cette section, nous montrons notre démarche de spécification de processus d'affaires telle que présentée dans la section 3.3.3.

3.3.4.2 La décomposition en chaîne de valeur

La première étape de l'approche consiste à décomposer le processus d'affaires original en une chaîne de processus d'échange et de conversion. Cette décomposition suit le patron REA de conception de la chaîne de valeur de Pavel Hruby (2006). Ce patron sera présenté dans le chapitre 4.

Le processus d'approvisionnement est un échange REA élémentaire. Ce processus est non décomposable. La Figure 3.17 montre la chaîne de valeur du processus d'approvisionnement. Dans cette Figure, nous montrons les événements économiques Product Receipt et Cash Disbursement. L'événement Product Receipt permet d'incrémenter la quantité de la ressource Produit; alors que l'événement Cash Disbursement décrémente la ressource Cash.

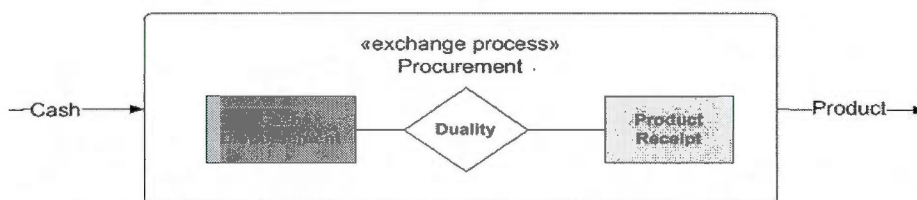


Figure 3.17 La chaîne de valeur du processus d'approvisionnement.

3.3.4.3 La représentation avec REA

Après la décomposition du processus d'affaires initial en une chaîne de processus d'échange et de conversion, nous allons modéliser chacun de ces processus. Dans notre exemple, nous avons un seul échange. Cette étape consiste à faire ressortir les vues REA, dynamique, informationnelle et organisationnelle. Nous avons remplacé la vue fonctionnelle de Curtis, Kellner et Over (1992) par une vue REA. En effet, cette dernière souligne les mêmes concepts que la vue fonctionnelle, tout en montrant les agents impliqués dans le processus. Dans le chapitre 5, nous montrons notre approche de représentation de processus d'affaires génériques avec plus de détails.

La représentation de la vue REA

La conception de la vue REA (l'équivalent de la vue fonctionnelle dans l'approche métalangage) utilise le patron de processus d'échange et le patron de processus de conversion de REA. Ces deux patrons seront présentés dans le chapitre 4. La vue REA doit considérer tous les événements (i.e. activités) du processus BPMN. Or, les patrons de base REA ne suffisent pas à modéliser les processus d'affaires de nos jours. En effet, REA traite seulement les événements économiques, alors que les processus d'affaires, de nos jours, sont de plus en plus complexes où davantage d'événements sont souvent impliqués dans un échange qui vont au-delà du simple *donner et recevoir* (*give and take*). Pour répondre aux besoins de ces processus, nous avons augmenté le patron REA de base pour inclure d'autres événements (i.e. activités) du processus BPMN. Cela rejoint les travaux de David (1997) qui a identifié d'autres types d'événements à considérer lors de l'analyse des systèmes avec le cadre REA. Nous présentons les travaux de David (1997) dans le chapitre 4.

Pour considérer tous les événements BPMN, l'analyste identifie les événements REA en se basant sur l'approche d'analyse de David (1997). Les événements économiques sont ceux qui changent la quantité de la ressource économique. À partir du diagramme BPMN, nous identifions les événements Product Receipt et Cash Disbursement. En effet, l'événement Product Receipt permet d'incrémenter la quantité de la ressource Produit; alors que l'événement Cash Disbursement décrémente la ressource Cash. Les autres événements offrent un support aux événements économiques tout en fournissant de nouvelles informations qui peuvent être utilisées par l'organisation pour prendre des décisions. En excluant les événements économiques qui sont des événements d'affaires spéciaux, nous identifions les événements suivants: Requisition Event, RFQ Event, Process RFQ, Select Supplier, PO Event, Process PO, Process Claim, et Materialize Claim. Nous allons voir dans les chapitres 4 et 5 que ce dernier événement (i.e. Materialize Claim) a une classification spéciale puisqu'il n'apporte pas de nouvelles informations pour l'organisation.

Au sein de notre méthodologie de représentation du processus générique, les événements économiques du partenaire d'affaires ne seront pas modélisés dans la vue REA étendue. En effet, cette stratégie permet d'éliminer la répétition puisque les événements économiques du même processus d'affaires vu par le partenaire d'affaires sont les mêmes que ceux de l'organisation pour laquelle nous modélisons (voir sect. 4.2.3). Ainsi, les événements Send Product et Payment Receipt sont les mêmes événements au sens de REA que les événements Product Receipt et Cash Disbursement respectivement. Nous appelons les événements Send Product et Payment Receipt *les événements miroirs* des événements Product Receipt et Cash Disbursement respectivement.

Le modèle final de la vue REA obtenue après l'augmentation par les événements non économiques forme *la vue REA étendue*. Cette vue présente les mêmes concepts qu'un sous-processus *Ad-hoc* selon la définition d'OMG pour BPMN. La Figure 3.18 montre la vue REA étendue du processus d'approvisionnement avec la notation entité-relation. La relation Duality relie les événements économiques duaux lors de l'échange. Ces derniers montrent quelle ressource a été augmentée (Product) et quelle ressource a été diminuée (Cash) en contrepartie. La relation Support représente le lien entre l'événement économique et l'autre

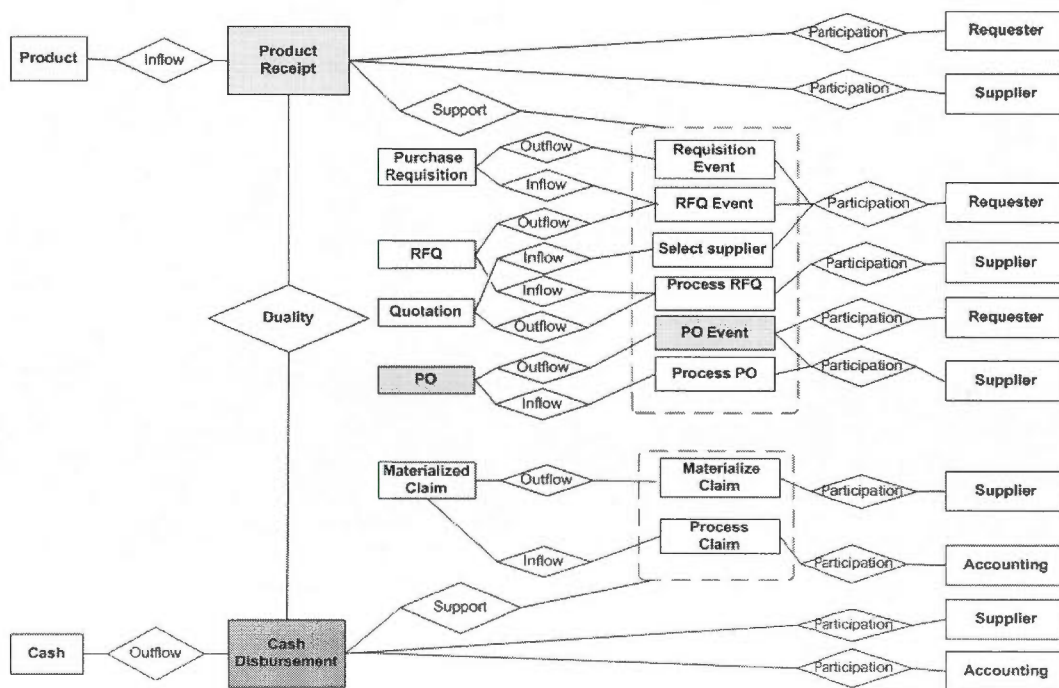


Figure 3.18 La vue REA étendue du processus d'approvisionnement.

événement REA qui le supporte (David, 1997). Les relations Inflow et Outflow relient une ressource à un événement REA. Finalement, la relation Participation décrit le lien entre l'agent et l'événement REA.

La représentation des vues organisationnelle, dynamique et informationnelle

Cette étape consiste à faire ressortir les vues dynamique, informationnelle et organisationnelle telles qu'identifiées par Curtis, Kellner et Over (1992). Pour des raisons de lisibilité, nous utiliserons le plus possible les concepts graphiques de BPMN pour illustrer la vue dynamique et la vue organisationnelle. Aussi, nous montrons *les événements économiques miroirs* pour faciliter l'interprétation de ces deux vues. Lors de la représentation de la vue dynamique et la vue organisationnelle, les éléments qui n'en font pas partie, mais qui y sont référencés, sont représentés en lignes pointillées. Dans ce cas, la vue représentée fait référence à une autre vue du processus. La Figure 3.19 montre la vue organisationnelle simplifiée. Les événements montrés font référence à la vue REA. La vue dynamique modélise l'ordonnancement des événements REA ainsi que les éléments de contrôle qui les

connectent. En BPMN, la vue dynamique est représentée par les séquences, les messages, les événements et les passerelles. Ces éléments sont illustrés en gras foncé dans la Figure 3.20.

Pour montrer notre approche, nous simplifions la vue organisationnelle. Nous montrons essentiellement les rôles, le type d'agent ainsi que les fonctions des agents (i.e. les événements REA dont l'agent est responsable).

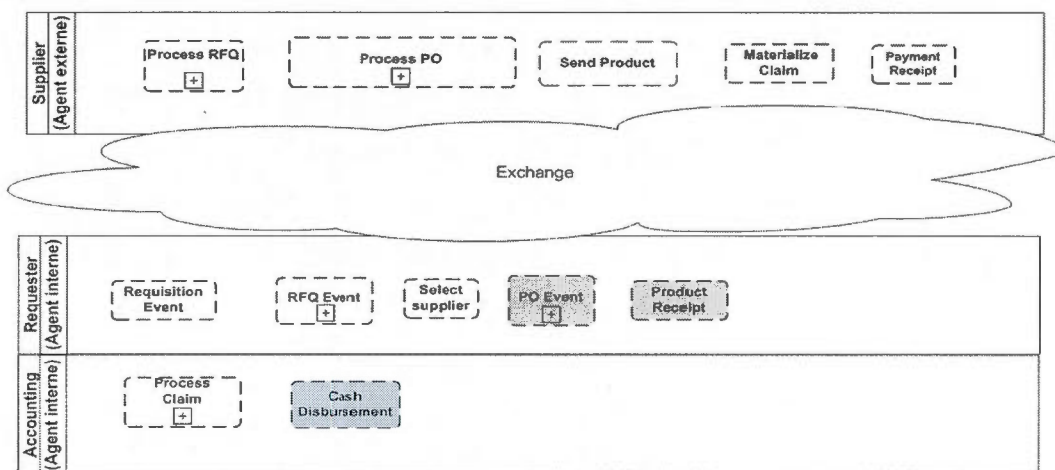


Figure 3.19 La vue organisationnelle du processus d'approvisionnement.

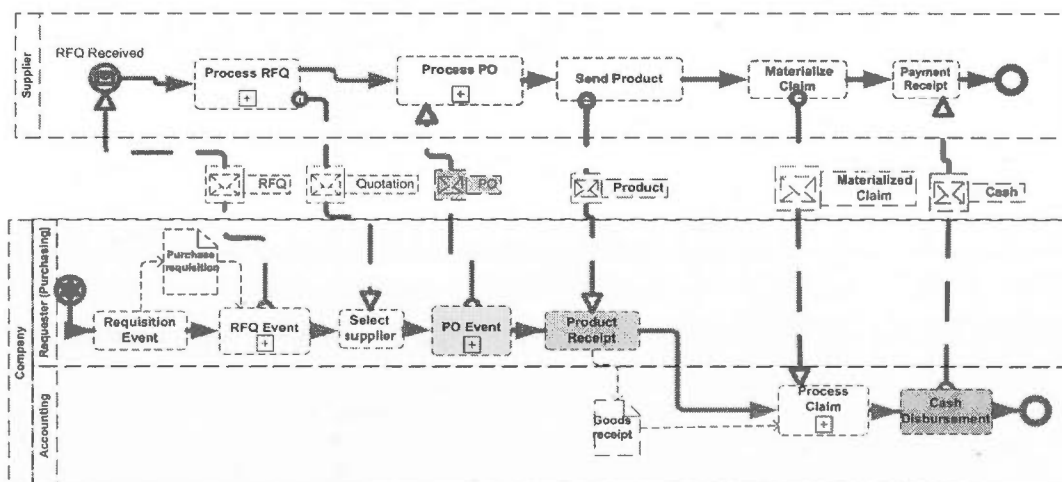


Figure 3.20 La vue dynamique du processus d'approvisionnement.

La vue informationnelle du processus d'approvisionnement (fig. 3.21) montre essentiellement la structure des ressources. Le bon de commande (PO) représente le contrat REA entre les agents économiques. Les lignes du contrat (PurchaseItem et PaymentLine) sont les obligations futures du contrat. Un contrat peut avoir plusieurs lignes d'achat et plusieurs lignes de paiement. En général, les obligations REA sont liées au type de la ressource et non à la ressource REA elle-même. L'entité EClaim représente la créance économique. Cette dernière décrit le déséquilibre au niveau des ressources économiques échangées durant le laps de temps qui sépare les événements économiques duaux. Dans notre cas, la créance économique est matérialisée par une facture (EMaterializedClaim). Pour faciliter la lecture et la compréhension de la vue informationnelle, nous avons décidé de représenter les agents REA impliqués dans le processus d'affaires.

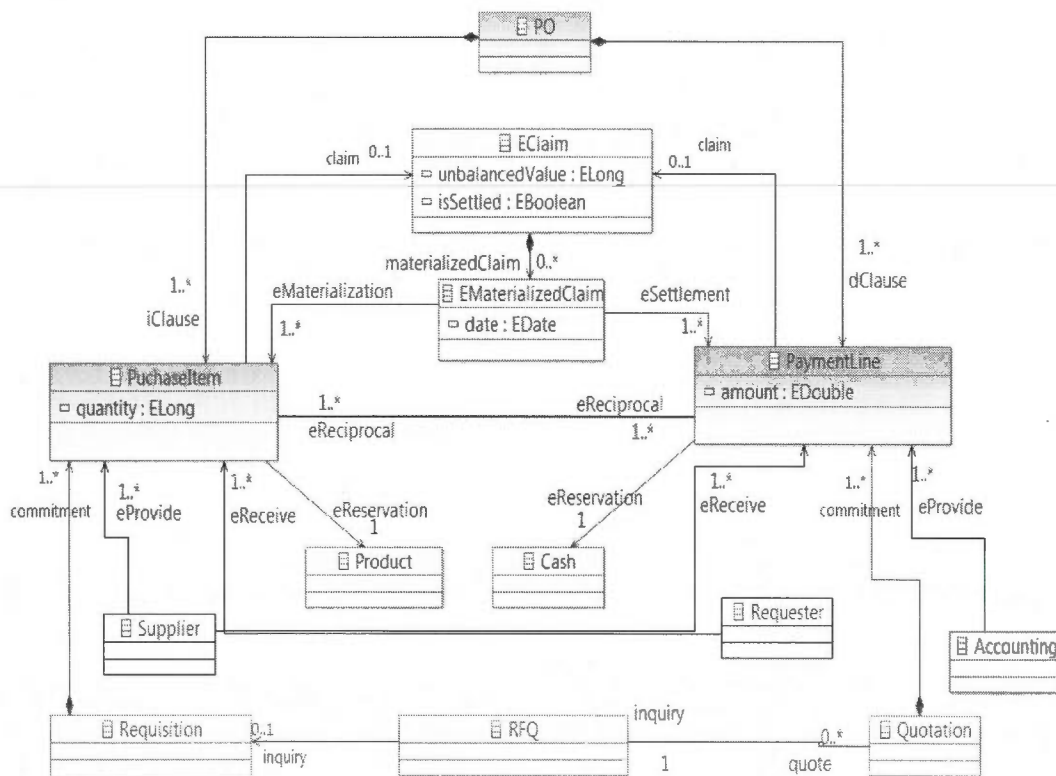


Figure 3.21 La vue informationnelle du processus d'approvisionnement.

3.3.4.4 La spécialisation du processus d'approvisionnement avec REA

Comme l'approche métalangage, la spécialisation consiste à appliquer une série de transformations génériques à un processus d'affaires. Plus précisément, les transformations sont appliquées aux quatre vues du processus d'affaires, soit, les vues REA, dynamique, informationnelle et organisationnelle.

À partir des patrons d'affaires REA, nous avons développé des transformations génériques. Par exemple, pour la question du contrat, voici les transformations génériques à appliquer aux vues d'un processus d'affaires dans le cas d'une réponse positive.

- *Vue REA* : Supprimer les événements REA liés à l'identification de l'agent partenaire. Les ressources utilisées exclusivement par ces événements sont retirées de la vue REA.
- *Vue Organisationnelle* : Supprimer le lien entre les agents et les événements REA liés à l'identification de l'agent partenaire.
- *Vue Dynamique* : Ajuster les liens comportementaux suite à la suppression des événements REA liés à l'identification de l'agent partenaire. Cela se traduit par supprimer/modifier/ajouter des séquences d'ordonnancement et de messages dépendamment des événements REA supprimés.
- *Vue Informationnelle*:
 1. Ajouter une nouvelle classe *Agreement* pour représenter l'accord.
 2. Ajouter les associations bidirectionnelles entre cette classe et les classes qui représentent les agents économiques selon le patron d'affaires *Contrat*.
 3. Ajouter les associations bidirectionnelles entre cette nouvelle classe et l'entité qui modélise le contrat REA selon le patron d'affaires *Contrat*.
 4. Supprimer de la vue tous les éléments informationnels liés aux événements d'affaires REA d'identification des agents partenaires.

Dans notre exemple du processus d'approvisionnement, l'existence d'un accord entre l'acheteur et le fournisseur éliminera le besoin de rechercher un fournisseur en fonction

du produit demandé car le fournisseur est déjà connu. Au niveau du modèle, ceci va se manifester de la manière suivante :

- Dans la vue REA, nous devons enlever les événements d'affaires d'identification du fournisseur, soit RFQ Event, Process RFQ, Select Supplier. Les ressources RFQ et Quotation reliées, exclusivement à ces événements, seront aussi retirées de la vue.
- Dans la vue organisationnelle, les événements supprimés depuis la vue REA seront retirés des responsabilités des agents.
- Dans la vue dynamique, nous devons ajuster les associations et les messages reliés aux événements RFQ Event, Process RFQ, Select Supplier. Cela se traduit par supprimer/modifier/ajouter les séquences d'ordonnancement et les messages reliés à ces événements.
- Dans la vue informationnelle, nous devons ajouter une nouvelle classe Agreement et des associations bidirectionnelles la reliant à l'acheteur, au fournisseur et à l'ordre d'achat selon le patron d'affaires Contract (fig. 3.16). Nous devons par la suite supprimer de la vue, toutes les ressources liés à d'identification du fournisseur (i.e RFQ et Quotation),

Les Figures 3.22, 3.23, 3.24 et 3.25 montrent le résultat d'application de la règle de transformation du patron *Contrat* aux vues REA, organisationnelle, dynamique, et informationnelle respectivement. Les éléments supprimés des vues REA, organisationnelle et dynamique sont identifiés par une croix. Nous rappelons que les éléments de la vue dynamique sont illustrés en gras foncé dans la Figure 3.24.

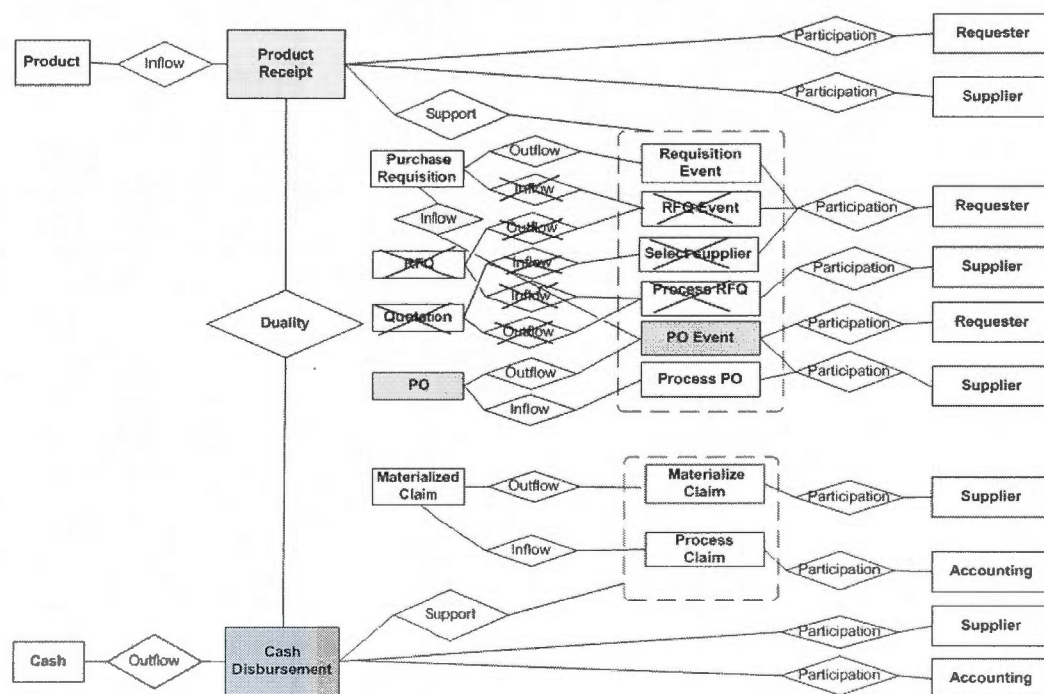


Figure 3.22 La vue REA après la transformation.

Dans la vue organisationnelle, les événements RFQ Event, Process RFQ et Select Supplier, supprimés depuis la vue REA, seront retirés des responsabilités des agents économiques.

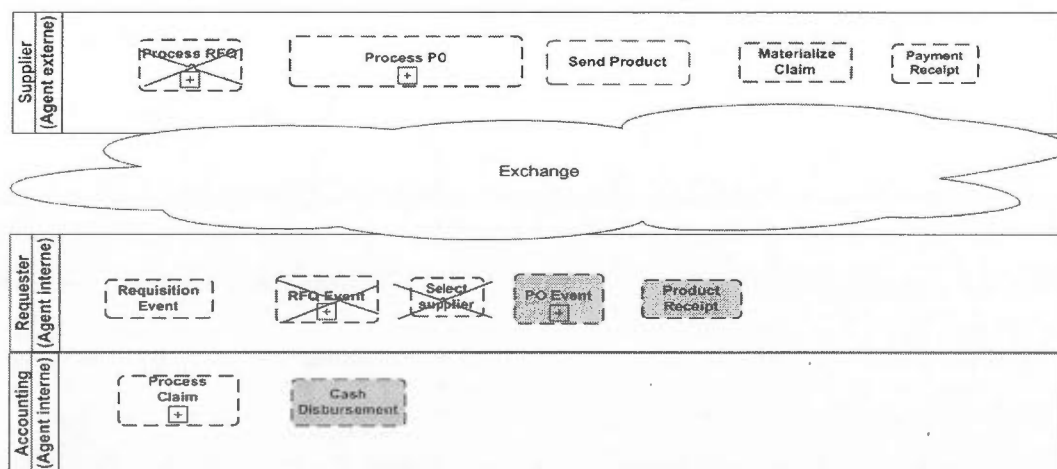


Figure 3.23 La vue organisationnelle après la transformation.

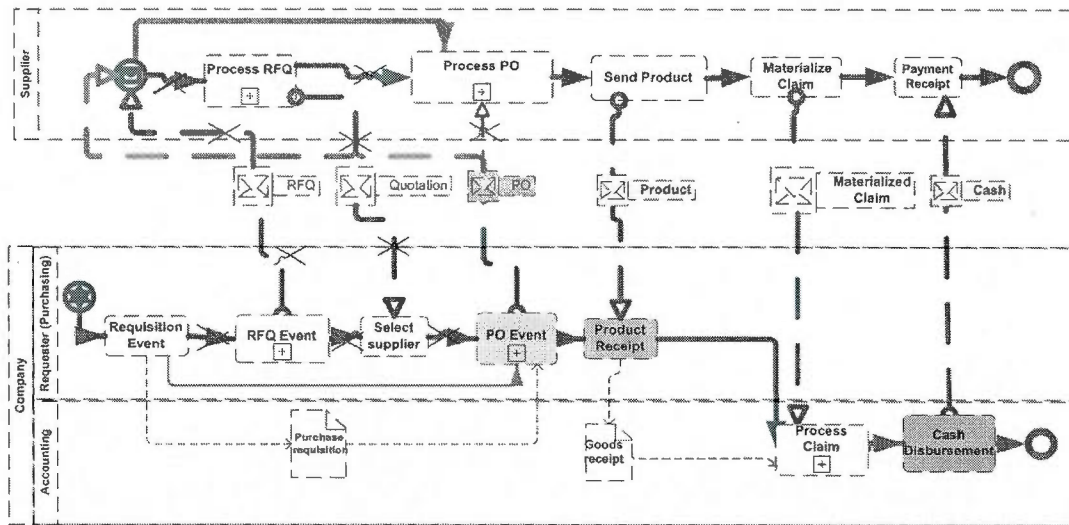


Figure 3.24 La vue dynamique après la transformation.

Dans la vue dynamique (fig. 3.24), nous devons enlever les séquences et les messages liés aux événements RFQ Event, Process RFQ et Select Supplier. Par exemple, les messages d'envoi des ressources RFQ et Quotation seront supprimés. Nous devons ensuite modifier/ajouter des séquences d'ordonnancement dépendamment des événements REA supprimés. Par exemple, la séquence entre l'événement Requisition Event et l'événement RFQ Event du contrat sera remplacée par une nouvelle séquence entre Requisition Event et PO Event.

Finalement, dans la vue informationnelle, nous devons ajouter une nouvelle classe Agreement. Nous devons aussi ajouter des associations bidirectionnelles la reliant aux agents économiques REA (i.e. l'acheteur et le fournisseur) et au contrat REA (i.e. l'ordre d'achat). Ces modifications sont dirigées par le patron d'affaires Contrat. Nous devons par la suite supprimer de la vue toutes les ressources liées aux événements d'affaires d'identification du fournisseur (i.e RFQ et Quotation).

Le processus obtenu sera ajouté dans notre catalogue commun de processus d'affaires. La Figure 3.26 illustre le processus d'approvisionnement final en BPMN obtenu après la transformation des quatre vues.

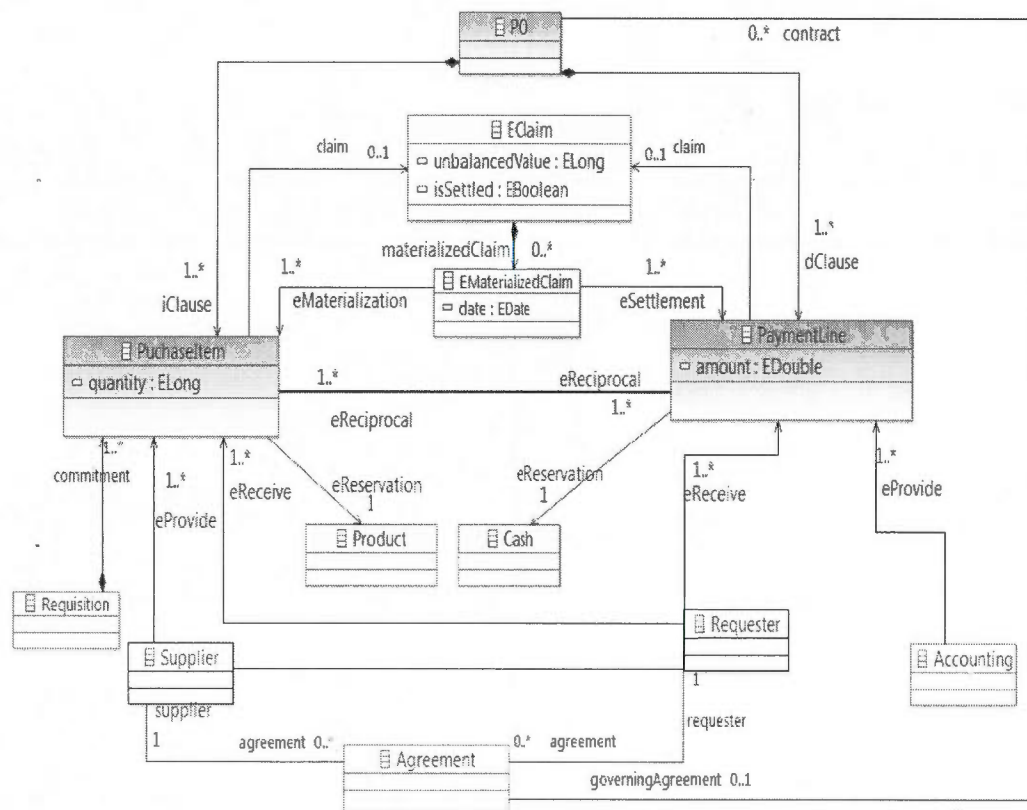


Figure 3.25 La vue informationnelle après la transformation.

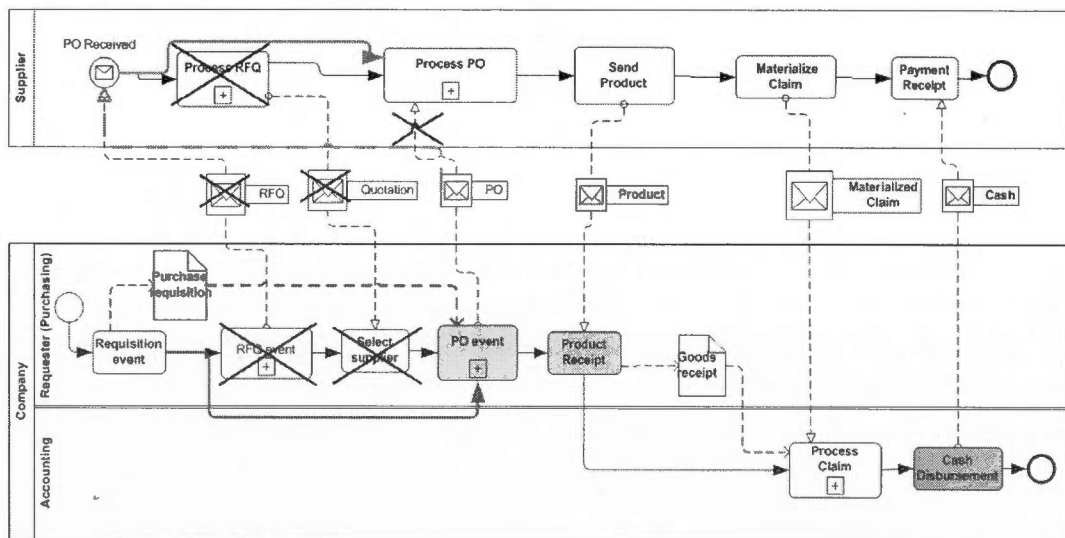


Figure 3.26 Le processus d'approvisionnement final après la transformation.

3.4 Notre approche de validation

Pour expérimenter notre approche, nous l'avons appliquée à différents processus d'affaires provenant de différents domaines d'affaires. Pour cela, nous avons considéré principalement la chaîne de valeur de Michael Porter (1975) et les processus clés des systèmes PGI (Magal et Word, 2010).

Pour valider notre approche, nous avons identifié un ensemble d'aspects qui doivent être vérifiés. Nous avons identifié les aspects suivants :

1. Que l'on puisse représenter les processus dans chaque domaine.
2. Que les questions génériques identifiées s'appliquent bien aux différents domaines.
3. Que les transformations élucidées s'appliquent aussi bien à tous ces processus d'affaires.

Nous présentons au chapitre 7 les résultats de la validation de notre approche. En particulier, nous expliquerons la démarche que nous avons adoptée pour valider les aspects mentionnés ci-haut, soit la représentation des processus d'affaires de différents domaines, l'applicabilité des questions et la correction de nos transformations de spécialisation. Nous présentons également les résultats de cette validation. Nous discutons aussi des problèmes que nous avons rencontrés quant à l'ordre des transformations chevauchantes.

3.5 Conclusion

Nous avons introduit, dans ce chapitre, une nouvelle approche pour la réutilisation de modèles de processus d'affaires. Notre méthodologie utilise une approche transformationnelle basée sur des questions. Elle a les caractéristiques suivantes : (i) elle est facile à assimiler par des utilisateurs ayant peu de connaissances techniques, (ii) le catalogue ne requiert pas d'être exhaustif, (iii) la représentation des processus d'affaires au sein du catalogue est basée sur une ontologie d'affaires, (iv) les questions, comme les transformations sont basés sur des patrons d'affaires, et (v) elle est générique car elle se base

sur des questions et des transformations qui s'appliquent à des processus d'affaires provenant de différents domaines.

Pour réutiliser un processus d'affaires, il faut d'abord le localiser. Pour cela, nous proposons la construction d'un catalogue de processus d'affaires génériques. Ensuite, nous avons montré comment modéliser le processus d'affaires générique avec le cadre ontologique REA. Nous avons proposé par la suite, une approche transformationnelle basée sur le concept de question/réponse. Cette dernière permet de spécialiser les vues du processus d'affaires, soit les vues REA, dynamique, informationnelle et organisationnelle.

CHAPITRE IV

L'ONTOLOGIE D'AFFAIRES REA

Nous avons présenté, dans le chapitre 3, une vue globale de notre approche de réutilisation de processus d'affaires. Nous avons exposé l'avantage d'utiliser une ontologie d'affaires pour la représentation et la spécialisation de processus d'affaires. Aussi, nous avons présenté l'ontologie d'affaires REA (Geerts et McCarthy, 2000). Cette dernière décrit le concept de création et d'échange de ressources économiques qui, de plus, est devenue le fondement de plusieurs normes d'échange électronique.

Ce chapitre sera dédié à l'ontologie d'affaires REA. La section 4.1 décrit notre motivation quant au choix d'une ontologie d'affaires, en l'occurrence REA. Nous présentons dans cette section les concepts de base de REA. La section 4.2 sera consacrée à la modélisation de processus d'affaires avec REA. Nous y présentons les patrons d'affaires d'échange et de conversion. La section 4.3 présente des extensions ontologiques de REA.

4.1 Motivation pour l'approche ontologique

Nous visons une approche de spécification de processus d'affaires basée sur les questions. Cette approche doit être facile à assimiler par des utilisateurs ayant peu de connaissances techniques. Les questions sont utilisées pour transformer un processus d'affaires générique en un processus d'affaires spécifique à une organisation. Notre approche se veut générique de sorte que les questions et les transformations puissent s'appliquer à des processus d'affaires de différents domaines. Si ces questions ont un sens métier, les transformations seront plus faciles

à concevoir. De plus, ces questions auront un sens pour les utilisateurs ayant peu de connaissances techniques tels qu'un analyste d'affaires.

Nous avons présenté, dans le chapitre 3 (voir sect. 3.2.7), une première approche à base d'un méta-modèle traditionnel de processus d'affaires. Notre validation a montré que les questions génériques, obtenues après la généralisation des questions spécifiques, ont perdu la sémantique d'affaires. Il en résultait deux inconvénients majeurs: (i) d'abord, la méthode génère beaucoup de questions superflues, et (ii) il est très difficile de trouver des transformations génériques qui ont du sens pour différents groupes de processus d'affaires. Nous avons montré que ce problème est en relation avec le concept de "généralisation". En effet, l'analyse de notre première approche montre que le passage méta vers le méta-modèle du processus d'affaires perd la sémantique d'affaires. Concrètement, il faut essayer de trouver à quel type de processus d'affaires appartiennent les processus étudiés. Par la suite, il faut identifier des points communs pour différents groupes de processus d'affaires.

Nous avons montré qu'il est plus intéressant de considérer les processus d'affaires comme des processus où des ressources sont échangées plutôt que de se concentrer, par exemple sur les différents acteurs dans un processus. De là, nous avons cherché une ontologie du domaine de processus d'affaires nous permettant de mettre en évidence les similitudes entre les différents processus. Cette ontologie nous permettra de modéliser nos processus d'affaires, de développer notre catalogue de questions génériques, et de mettre en œuvre des spécialisations qui ont du sens métier. Selon Ciuksys et Caplinskas (2007), l'utilisation d'une ontologie d'affaires permet la réutilisation des connaissances métier et l'adaptation des processus d'affaires aux besoins des entreprises et non le contraire.

La solution ontologique présente l'avantage d'éliminer les questions spécifiques superflues générées lors de l'instanciation dans la première approche. En effet, prenons l'exemple de la question générique du contrat. Cette question a été formulée comme suit dans la première approche : *“Est-ce que le déroulement du processus est régi par un contrat entre <Acteur> et <Acteur> ?”* où <Acteur> est un élément du méta-modèle représentant un acteur.

Dans notre exemple de processus d'approvisionnement présenté dans le chapitre 3 (fig. 3.16), nous retrouvons trois acteurs au niveau organisationnel: « Requirer », « Supplier » et « Accounting ». Ainsi, notre question générique aura autant d'instanciations qu'il y a de

combinaisons possibles entre ces trois paramètres. Pour nos questions spécifiques, nous obtenons l'ensemble des paires suivantes: (« Requester », « Requester »), (« Requester », « Supplier »), (« Requester », « Accounting »), (« Supplier », « Requester »), (« Supplier », « Supplier »), (« Supplier », « Accounting »), (« Accounting », « Requester »), (« Accounting », « Supplier »), et (« Accounting », « Accounting »). Plusieurs questions spécifiques associées à ces combinaisons sont clairement non pertinentes. De plus, voici quelques-unes des difficultés rencontrées avec deux heuristiques :

1. Il faut considérer seulement les acteurs qui interagissent directement. Cela peut sembler raisonnable à première vue, mais dans un contexte organisationnel, les personnes qui signent le contrat ne sont pas celles qui interagissent dans le processus d'échange.
2. Il faut considérer seulement les acteurs qui appartiennent à différentes unités organisationnelles. Ceci s'est avéré ambigu. Dans une organisation, les interactions entre les unités d'affaires ou même entre des départements au sein d'une même unité d'affaires peuvent être régies par des accords ou des contrats de service. Par exemple, dans des sociétés où le département informatique est géré comme un centre de profit, les services informatiques sont facturés à différents départements.

Avec l'ontologie REA, tous les processus d'affaires sont considérés comme des séquences d'échanges. À partir de cette constatation, il est plus facile de voir que la notion de contrat devient pertinente dans tous les domaines de processus comme un moyen de gouverner ces échanges d'affaires. La question du contrat peut alors être formulée comme suit : *Est-ce que l'échange est régi par un accord ?* Avec cette nouvelle question, les deux heuristiques précédentes sont directement intégrées. En effet, 1) les agents impliqués interagissent directement dans l'échange, et 2) ils appartiennent à deux unités organisationnelles, ce qui est implicite dans l'ontologie REA.

4.1.1 Le choix de l'ontologie d'affaires REA

Une ontologie est la spécification d'une conceptualisation d'un domaine de connaissances (Gruber, 1993). Une ontologie de domaine est l'ensemble structuré de concepts représentant le

sens des éléments dans un domaine ainsi que les relations entre ces concepts. Elle permet d'améliorer la communication, le partage et la réutilisation (Ushold et Gruninger, 1996). Une ontologie d'affaires est une ontologie qui souligne principalement le concept d'échange et de transformation de ressources économiques.

Principalement, il existe deux types d'ontologies dans le domaine d'affaires (1) les ontologies d'entreprise, et (2) les ontologies d'affaires. L'objectif des ontologies d'entreprise, comme TOVE (*Toronto Virtual Enterprise*) (Fox, 1992) et (Ushold *et al.*, 1998), est de souligner les activités d'une entreprise. Principalement, elles visent à créer des représentations réutilisables des connaissances internes à une entreprise, par exemple, les activités liées à la gestion organisationnelle, la structure, etc. Ces ontologies ne sont pas axées sur le transfert de valeur économique entre les acteurs économiques (Ilayperuma, 2007). Les ontologies d'affaires, quant à elles, sont dirigées vers la création et l'échange de ressources économiques. Elles sont axées principalement sur la collaboration entre les partenaires d'affaires dans le contexte d'échange et de transformation de ressources économiques.

Nous avons le choix entre trois ontologies d'affaires, REA (Geerts et McCarthy, 2000), l'ontologie e³-value (Gordijn et Akkermans, 2001) et e-BMO (Osterwalder et Pigneur, 2002). Quelques travaux ont essayé d'unifier ces ontologies d'affaires (Andersson *et al.*, 2006), ou de proposer une ontologie d'affaires de référence (Ilayperuma, 2007) ou encore de créer des correspondances entre elles (Schuster et Motal, 2009). En l'absence d'une ontologie acceptée, nous devons en choisir une pour illustrer notre approche. Nous avons choisi l'ontologie REA puisqu'elle est simple et qu'elle couvre le concept d'échange et de transformation de ressources économiques. Selon Gailly et Poels (2007), les modèles REA sont plus proches des modèles d'affaires de e³-value et e-BMO. De plus, l'ontologie REA est devenue le fondement de plusieurs normes d'échange électronique, telles que ebXML, ISO Open-EDI, UMM (*UN/CEFACT Modelling Methodology*) et d'autres initiatives telles que le méta-framework ECIMF (*Electronic Commerce Integration Meta Framework*) pour l'interopérabilité en commerce électronique. Finalement, REA fait partie d'une nouvelle norme OMG en cours, appelée VDML (*Value Delivery Modeling Language*). En 2011, cette norme était dans sa première demande de soumission. Elle propose un méta-modèle pour la gestion et

l'échange de valeurs économiques entre l'organisation, ses partenaires et ses clients. La création et l'échange, qui sont les patrons de base de REA, représentent le cœur de VDML.

Hruby (2006) présente REA comme une ontologie qui permet de spécifier les règles fondamentales des domaines d'affaires, tout en assurant la cohérence des systèmes du point de vue métier. Voici quelques avantages de REA (Hruby, 2006)

1. Les applications conçues avec REA sont concises et faciles à comprendre;
2. REA permet la réutilisation des modèles d'affaires de différents domaines puisque les mêmes concepts sont décrits par les mêmes patrons;
3. Les systèmes basés sur REA contiennent plus de connaissances d'affaires par rapport à d'autres applications basées sur les exigences d'utilisateurs.

Selon David (1997), REA peut être un outil puissant dans d'analyse des processus d'affaires des organisations. Avec REA, les analystes et les gestionnaires se concentrent sur les éléments les plus fondamentaux en soulignant les activités qui consomment les ressources importantes tout en ajoutant de la valeur pour les clients. De plus, REA force les concepteurs à évaluer, de manière critique, les autres activités qui n'ajoutent pas directement de la valeur d'affaires (Porter, 1985). Enfin, Hruby (2006) décrit REA comme une ontologie qui offre une modélisation dirigée par les modèles qui révolutionnera le monde du développement des systèmes informatiques. Actuellement, un nombre grandissant d'analystes d'affaires affirment que les modèles qu'ils développent deviennent meilleurs quand ils considèrent REA lors de l'étape d'analyse et de conception (Hruby, 2006).

4.1.2 L'ontologie d'affaires REA

REA a été introduit par William E. McCarthy (1982) pour modéliser des systèmes d'information dans le domaine de la comptabilité. McCarthy (1982) propose REA comme un cadre qui permet de comptabiliser les phénomènes économiques dans un environnement inter-organisationnel.

REA se base sur le principe suivant : *Si une entreprise veut accroître la valeur totale d'une ressource sous son contrôle, elle doit, généralement, diminuer la valeur de certaines de*

ses ressources. REA perçoit un processus d'affaires en termes de création, de transformation et d'échange de valeurs économiques

REA offre une sémantique riche permettant la compréhension des processus d'affaires. Elle est devenue, par la suite une ontologie de modélisation des processus d'affaires. Le méta-modèle de base de REA est présenté à la Figure 4.1 avec la notation entité-relation. Les concepts de base de REA sont les ressources économiques, les événements économiques et les agents économiques.

- Les *ressources économiques* sont des objets rares et utiles qui sont sous le contrôle de l'organisation (McCarthy, 1982). Ce sont des objets que les gestionnaires des applications d'affaires veulent planifier, surveiller et contrôler. Par exemple, les produits, les services, les matières premières et la main-d'œuvre sont des ressources économiques. La valeur de la ressource économique est relative au détenteur des droits sur la ressource. Cette valeur n'est pas la même entre les agents économiques impliqués dans le processus REA.
- Les *événements économiques* sont définis comme étant des phénomènes qui reflètent des changements au niveau de la valeur des ressources économiques. Les ressources peuvent être incrémentées ou décrémentées dans le contexte d'un événement économique. Dans le cadre du processus d'affaires, tout incrément d'une ressource économique doit s'accompagner d'un décrement d'une autre ressource économique. C'est le principe de la dualité. La relation de dualité relie les événements économiques d'incrément (qui augmentent la valeur de la ressource) aux événements économiques de décrement (qui diminuent sa valeur). Elle permet de garder une trace de quelle ressource économique a été incrémenté, et quelle ressource économique a été décrementé.
- L'*agent économique* est un individu ou une organisation qui peut avoir un contrôle sur les ressources économiques.

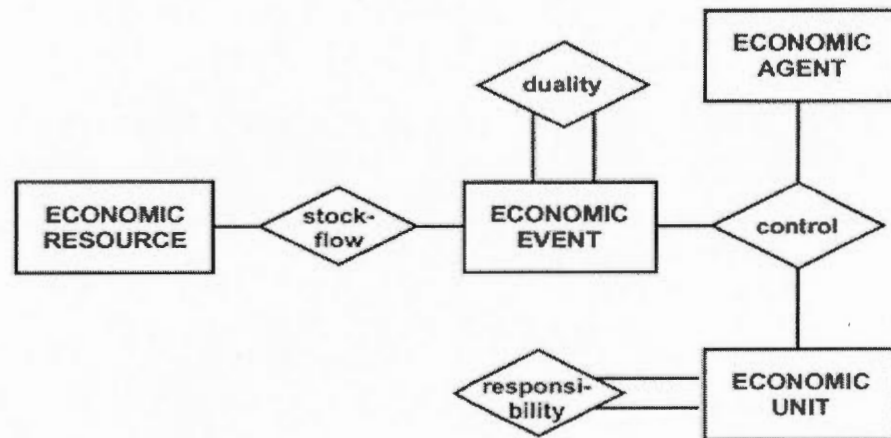


Figure 4.1 L'ontologie REA de base. (Source : McCarthy, 1982)

Geerts et McCarthy (2000) proposent trois axiomes de l'ontologie de base REA :

1. Pour chaque ressource économique, il existe au moins un événement d'incrément et un événement de décrément qui l'affecte. Inversement, un événement d'incrément ou de décrément doit affecter une ressource économique identifiable.
2. Tout événement effectuant un décrément doit être relié à un événement effectuant un incrément via la relation de dualité et vice versa.
3. Chaque échange implique des agents économiques. Un agent peut être interne ou externe à l'organisation.

4.1.2.1 Exemple d'échange économique

L'exemple suivant illustre un processus d'échange élémentaire en REA entre le client qui est *Cookie-Monster*, et l'entrepreneur *Elmo* (McCarthy, 2004). *Cookie-Monster* est un agent économique qui possède la ressource économique *Cash*. L'entrepreneur *Elmo* est un agent économique qui possède la ressource économique *Cookie*. La Figure 4.2 montre le client *Cookie-Monster* et l'entrepreneur *Elmo* avant de s'engager dans le processus d'échange économique REA.



Figure 4.2 La situation économique avant l'échange économique. (Source : McCarthy, 2004)

Cookie-Monster veut acheter un biscuit de l'entrepreneur *Elmo*. D'abord, *Cookie-Monster* et *Elmo* s'engagent dans le processus de vente du biscuit (i.e. la fourniture de la ressource économique *Cookie*) qui consiste en un transfert de la ressource *Cookie* d'*Elmo* vers *Cookie-Monster*. Par la suite, les deux agents économiques s'engagent dans le processus de paiement qui consiste en un transfert du *Cash* de *Cookie-Monster* vers *Elmo*. La vente qui consiste en le transfert de **droit de propriété** de la ressource *Cookie* est un événement économique. La Figure 4.3 montre la situation économique après la vente.



Figure 4.3 La situation économique après le processus de vente. (Source : McCarthy, 2004)

À ce niveau, la vente qui est la première partie de l'échange économique est complétée. Le client *Cookie-Monster* a reçu la ressource *Cookie* de l'entrepreneur *Elmo*. À ce moment là, *Elmo* n'a pas encore reçu l'argent de *Cookie-Monster*. La Figure 4.4 montre le modèle partiel de la vente. Le modèle est conçu du point de vue de l'entrepreneur *Elmo*.

Par la suite, *Cookie-Monster* et *Elmo* s'engagent dans le processus de paiement. Le paiement est un événement économique qui consiste en un transfert de **droit de propriété** de la ressource économique *Cash* de *Cookie-Monster* vers l'entrepreneur *Elmo*. C'est l'événement économique **dual** de la vente. La Figure 4.5 montre la situation économique après l'événement de paiement.

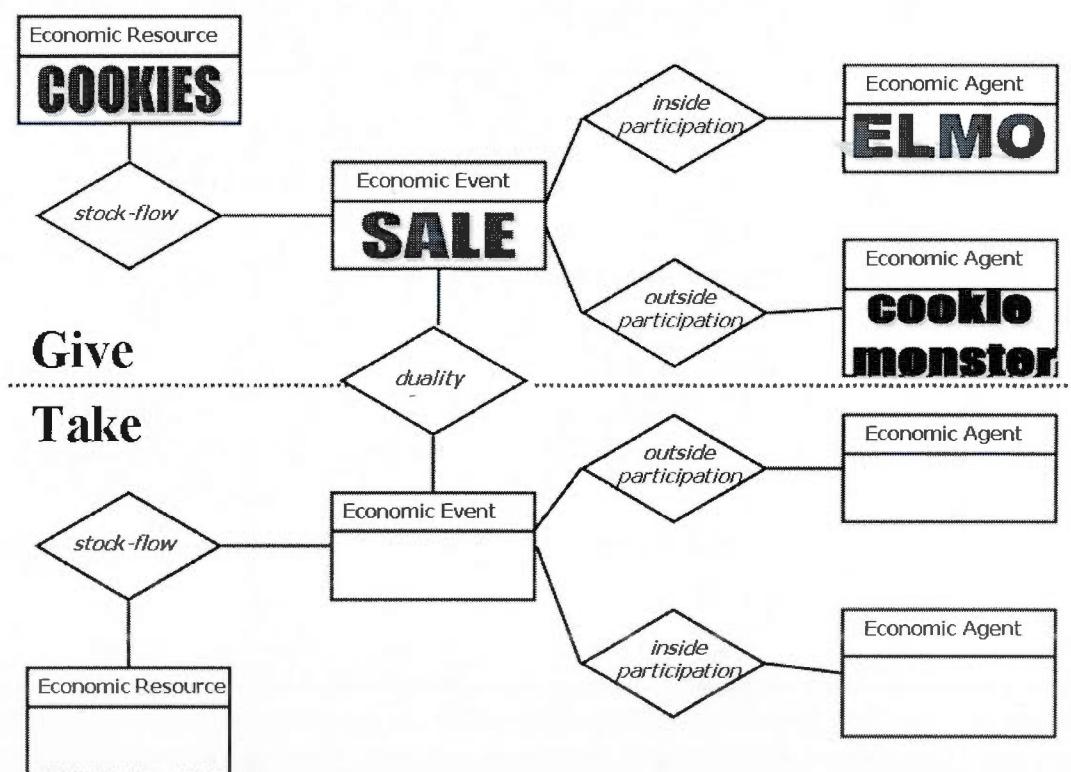


Figure 4.4 Le modèle REA de la vente vu par Elmo. (Source : McCarthy, 2004)

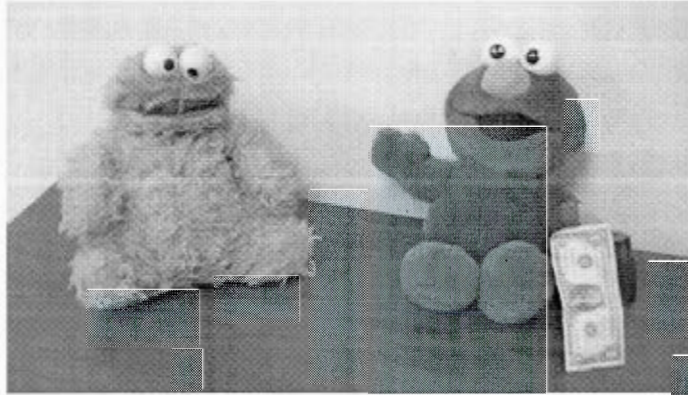


Figure 4.5 La situation économique après le processus de paiement. (Source : McCarthy, 2004)

Le paiement complète le processus d'échange économique. La Figure 4.6 montre le modèle REA complet du processus d'échange (conçu de la perspective de l'entrepreneur *Elmo*). Comme nous pouvons le constater, l'échange consiste en un événement d'incrément (i.e. Réception du cash) et d'un événement de décrétement (i.e. Vente). Ces événements sont associés par une relation de dualité d'échange. Cette relation permet de garder une trace des ressources économiques échangées. Chaque événement d'incrément peut être relié à un ou plusieurs événements de décrétement et inversement.

Pour que le processus d'échange ajoute de la valeur économique, le total des incréments (en termes de valeur des ressources économiques associées aux événements d'incrément) doit être supérieur au total des décréments (en termes de valeur des ressources économiques associées aux événements de décrétement).

Comme le montre la Figure 4.6, chacun des événements économiques (Vente et Réception du cash) est relié aux deux agents économiques *Elmo* et *Cookie-Monster*. En règle générale, chaque événement est relié à exactement deux agents : celui qui fournit la ressource et celui qui reçoit la ressource. Notez, cependant, que l'agent *fournisseur* d'un événement n'est pas obligatoirement l'agent *receveur* de son événement dual.

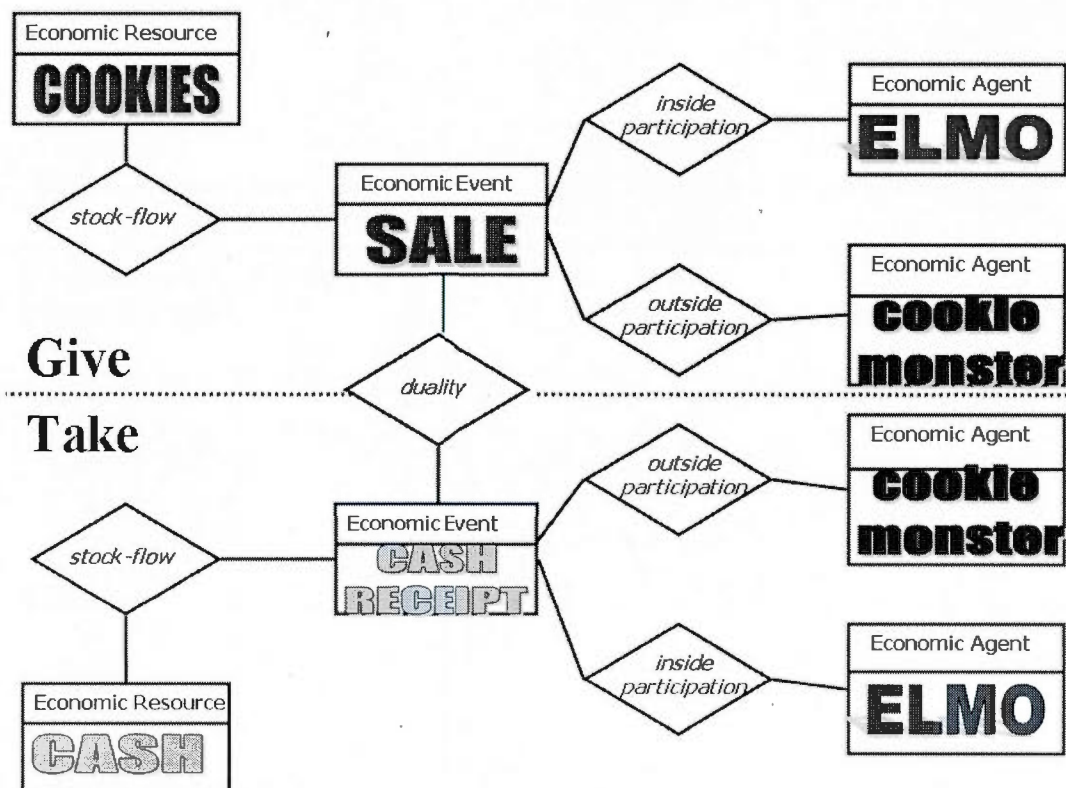


Figure 4.6 Le modèle REA complet de l'échange vu par Elmo. (Source : McCarthy, 2004)

Les modèles REA se basent par défaut sur la perspective de l'entreprise pour laquelle on veut modéliser le processus d'affaires. Dans ce cas, l'agent interne est l'entreprise (i.e. *Elmo*) et l'agent externe est le partenaire (i.e. *Cookie-Monster*). La perspective du partenaire pour le même processus n'est qu'une vue miroir de celle de l'entreprise. Il existe aussi la perspective d'un point de vue indépendant. Dans ce cas, on doit montrer clairement la source et la destination de la ressource économique. La Figure 4.7 montre le modèle du processus d'échange entre *Cookie-Monster* et *Elmo* du point de vue d'un agent indépendant.

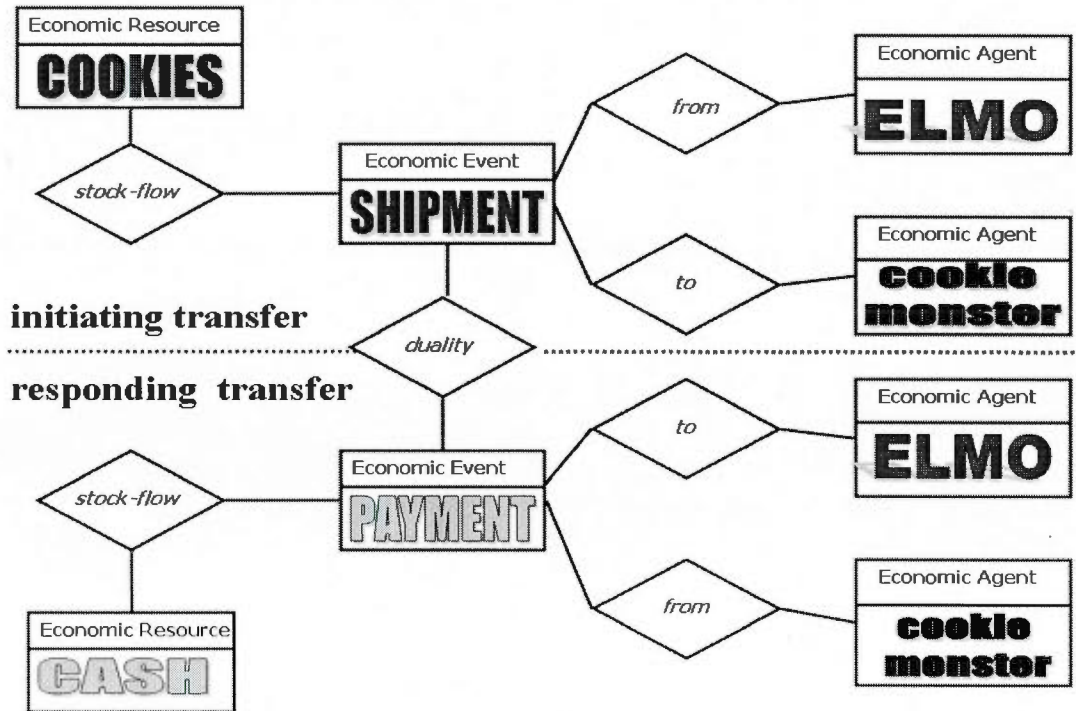


Figure 4.7 Le modèle REA du point de vue d'un agent indépendant. (Source : McCarthy, 2004)

4.2 Modélisation de processus d'affaires avec REA

REA permet de modéliser deux types de processus : 1) les interactions entre l'entreprise et ses partenaires d'affaires dans le cadre d'échanges économiques, et 2) la création de nouvelles ressources ou la transformation de ressources existantes dans le cas de processus de conversion. La Figure 4.8 montre le méta-modèle de base en UML d'un processus d'affaires en REA.

La relation *linkage* permet de définir la relation entre les ressources. La composition est un exemple de *linkage*. La relation *association* décrit la dépendance entre les agents économiques. Geerts et McCarthy (2000) distinguent trois différents types de la relation *association*, soit : *responsibility*, *assignment*, et *cooperation*. Finalement, la relation *custody* décrit la relation de responsabilité d'un agent pour une ressource spécifique.

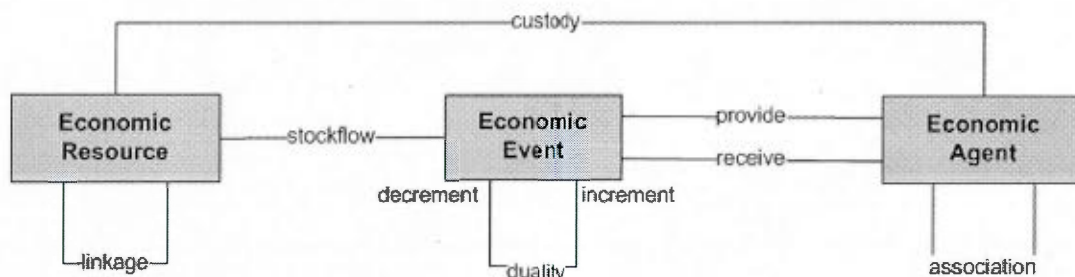


Figure 4.8 Le méta-modèle REA de base en UML. (Source : Geerts et McCarthy, 2000)

4.2.1 Le patron de processus d'échange REA

Le processus d'échange —de transfert dans sa version originale— consiste en un transfert de ressources économiques entre deux agents. En général, les agents sont l'entreprise et ses partenaires d'affaires. Mais il peut s'agir de deux agents au sein de la même entreprise. Pour que le processus ait un avantage économique pour l'organisation, la valeur ajoutée globale de la ressource reçue doit être supérieure à la valeur diminuée globale de la ressource fournie.

Dans un processus d'échange, la ressource économique est considérée comme un ensemble de droits détenus par les agents économiques. Les droits échangés peuvent être, par exemple, des droits à la propriété, des droits à l'utilisation, ou des droits à la copie. Le but principal d'un processus d'échange est de donner ou de recevoir des droits associés à une ressource économique tout en recevant ou en donnant les droits à d'autres ressources (Jaquet, 2006). Par exemple, lors de l'achat d'un appartement, l'entreprise reçoit le droit de propriété de l'appartement. Lors de la location de cet appartement, l'entreprise perd le droit d'utilisation pendant la période de la location. Finalement, lors de la vente, l'entreprise perd le droit de propriété de cet appartement. La Figure 4.9 montre le patron d'échange.

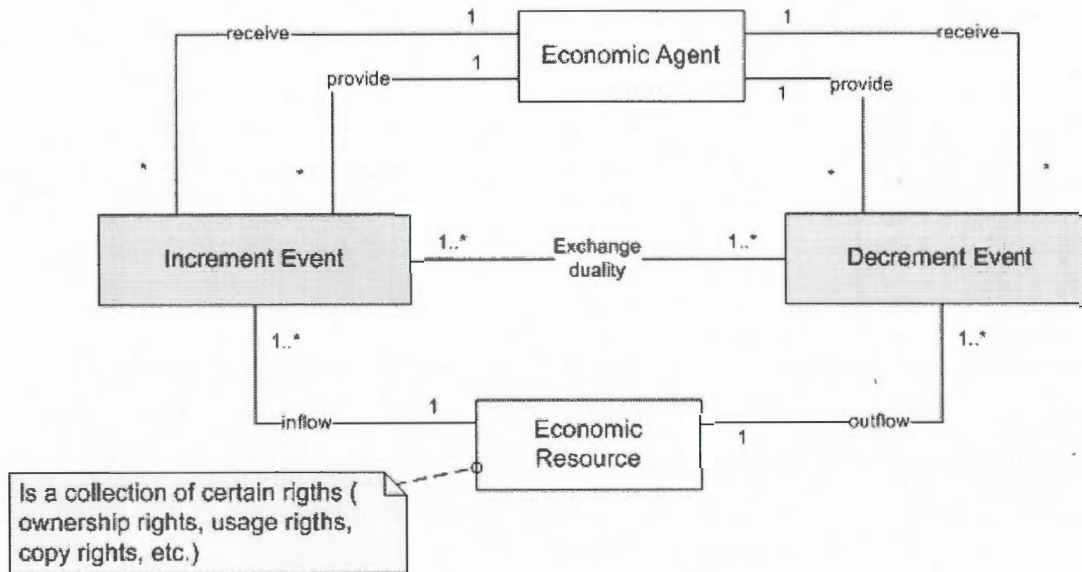


Figure 4.9 Le patron d'échange REA. (Source : Hruby, 2006)

Voici les règles du patron d'échange (Hruby, 2006):

- Chaque événement d'incrément (*increment event*) doit être relié à au moins un événement de décrétement (*decrement event*) par une dualité d'échange, et inversement.
- Chaque événement d'incrément doit être relié à une ressource économique par la relation *inflow*.
- Chaque événement de décrétement doit être relié à une ressource économique par la relation *outflow*.
- Chaque événement économique doit être relié à deux agents via les relations « provide » et « receive ».

L'événement économique dans un processus d'échange représente le transfert de droits associés à une ressource économique d'un agent vers un autre. Ce transfert peut être temporaire ou permanent. L'événement d'incrément augmente la valeur globale de la ressource qui lui est reliée. L'événement de décrétement diminue la valeur globale de la ressource. La

relation de dualité d'échange est une relation de plusieurs à plusieurs. L'événement d'incrément peut être relié à plusieurs événements de décrétement par une dualité d'échange et inversement. La Figure 4.10 montre un exemple de dualité d'échange liant un événement d'incrément à deux événements de décrétement.

La relation *inflow* relie une ressource économique à un événement d'incrément. Cela résulte par le transfert de droits sur des ressources vers l'entreprise. Par exemple, lors d'un achat, l'entreprise reçoit le droit de propriété de produits. La relation *outflow* relie une ressource économique à un événement de décrétement. Cela résulte en la perte de droits sur des ressources. Par exemple, suite à un achat, l'entreprise doit payer. Ainsi, elle perd le droit de propriété d'un montant d'argent.

Il est important de noter que lors d'un processus d'échange, les agents économiques liés aux événements d'incrément ne sont pas nécessairement ceux liés aux événements de décrétement. En effet, l'ontologie REA n'impose pas de relation de symétrie entre les agents par rapport à la relation de dualité. Ainsi, l'agent *fournisseur* d'un événement n'est pas obligatoirement l'agent *receveur* de son événement dual. Par exemple, le service d'achats peut être responsable de l'événement d'incrément (i.e. Achat), par contre l'événement de décrétement (i.e. Paiement) peut être fourni par le service des finances.

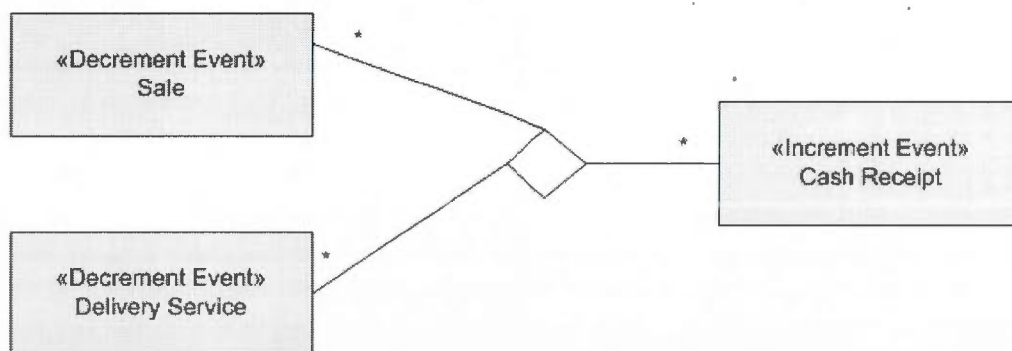


Figure 4.10 Un exemple de dualité d'échange. (Source : Hruby, 2006)

4.2.2 Le patron de processus de conversion REA

Le processus de conversion consiste en la création de nouvelles ressources économiques ou la transformation de ressources économiques existantes par *l'utilisation* ou *la consommation* d'autres ressources. Pour que le processus ait un avantage économique pour une organisation, la valeur globale ajoutée à la ressource par les événements de production (i.e. événements d'incrément) doit être supérieure à la valeur diminuée globale des ressources reliées aux événements d'utilisation ou de consommation (i.e. événements de décrétement). La Figure 4.11 montre le patron REA de processus de conversion.

Dans un processus de conversion, la ressource économique est considérée comme un ensemble d'attributs qui lui sont associés. Un attribut peut être une propriété, une caractéristique, une capacité ou un état qui détermine l'utilité d'une ressource pour un agent économique (Hruby, 2006).

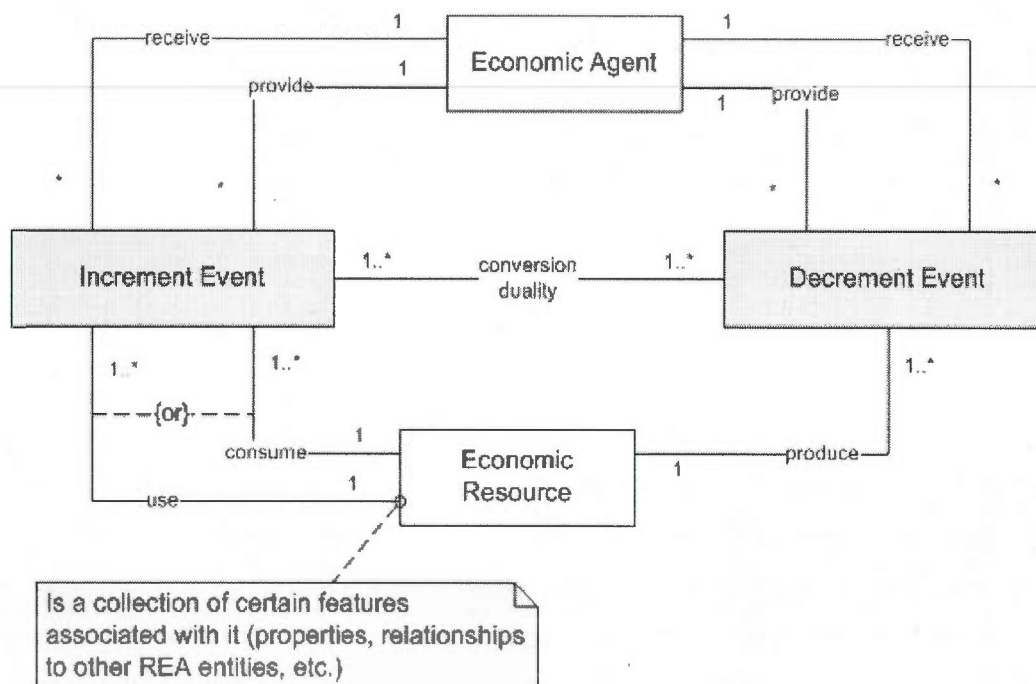


Figure 4.11 Le patron de conversion REA. (Source : Hruby, 2006)

Voici les règles du patron de conversion REA (Hruby, 2006):

- Chaque événement d'incrément doit être relié à au moins un événement de décrément par une relation de dualité de conversion, et inversement.
- Chaque événement d'incrément doit être relié à une ressource économique par la relation « produce ».
- Chaque événement de décrément doit être relié à une ressource économique par la relation « use » ou « consume ».
- Chaque événement économique doit être relié à deux agents via les relations « provide » et « receive ».

Dans un processus de conversion, on a des événements économiques pour la création d'une ressource économique, sa consommation ou le changement de (certains de) ses attributs. L'événement d'incrément augmente la valeur globale de la ressource qui lui est reliée par la relation « produce ». L'événement de décrément diminue la valeur globale de la ressource qui lui est reliée par la relation *use* ou *consume*.

La relation de dualité de conversion permet de relier les événements d'incrément aux événements de décrément. Elle permet de garder une trace de quelles ressources ont été utilisées ou consommées pour la production ou la transformation d'autres ressources. Comme la relation de dualité d'échange, la dualité de conversion est une relation de plusieurs à plusieurs. La Figure 4.12 montre un exemple de dualité de conversion liant un événement d'incrément à plusieurs événements de décrément.

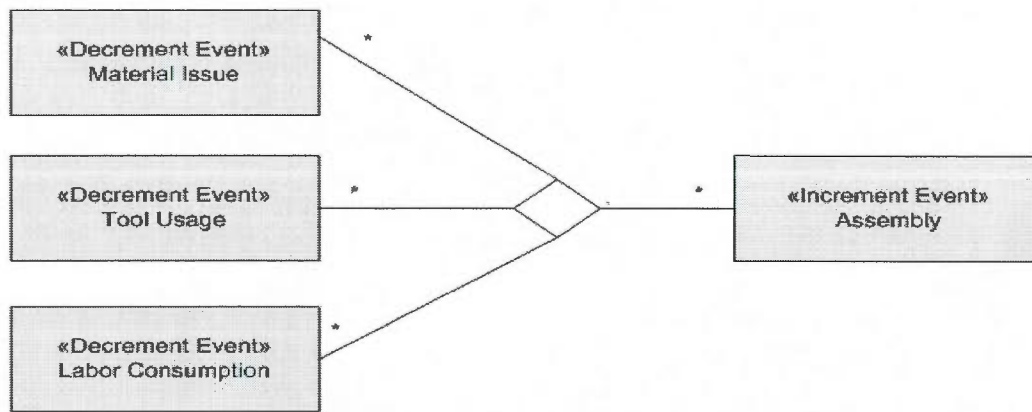


Figure 4.12 Un exemple de dualité de conversion.

4.2.3 La perspective REA de la modélisation des processus d'affaires

Notre approche de modélisation de processus d'affaires se base sur la perspective de l'entreprise pour laquelle nous modélisons le processus. Tous nos modèles suivent cette approche de représentation. Nous n'utiliserons pas la perspective du partenaire qui n'est qu'une vue miroir de celle de l'entreprise. Aussi, nous n'utiliserons pas la perspective d'un agent économique indépendant. La Figure 4.13 montre un exemple de modèles REA du point de vue de deux partenaires d'affaires, soit l'acheteur et le fournisseur. Le modèle du processus d'approvisionnement de la perspective de l'acheteur est l'image miroir du processus de vente du fournisseur. Dans cette Figure, nous appelons les événements économiques *Cash Disbursement* et *Product Receipt* les événements miroirs des événements économiques *Cash Receipt* et *Product Sales* respectivement.

Le fait d'utiliser la perspective du point de vue de l'entreprise ne représente pas une limite de notre approche. En effet, la modélisation REA du point de vue de l'entreprise facilite la lecture des modèles. Notre méthodologie de réutilisation et de transformation de modèles des processus d'affaires reste valide pour les autres perspectives de modélisation.

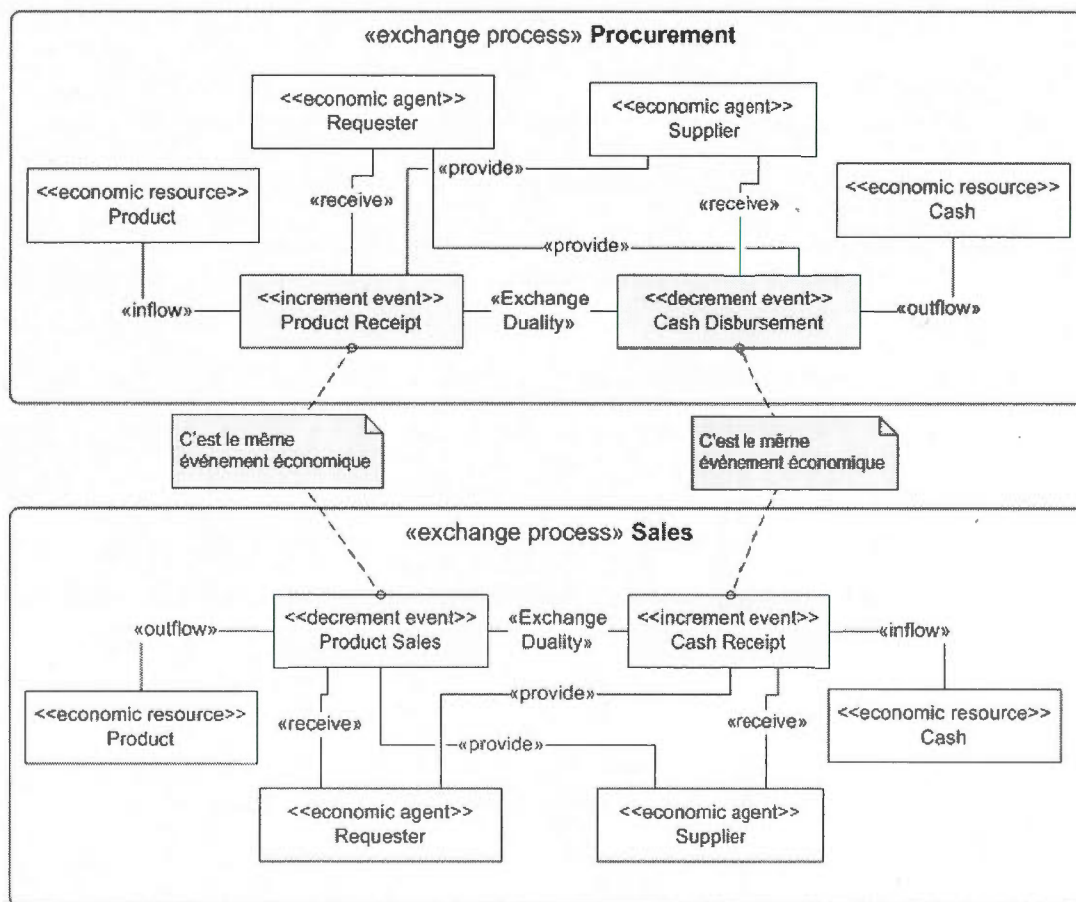


Figure 4.13 Le processus d'échange du point de vue des partenaires d'affaires.

4.2.4 La chaîne de valeur REA

Chaque entreprise crée de la valeur en fournissant des biens et des services. Ces derniers sont créés à travers une série de processus d'affaires planifiés, surveillés et contrôlés par différents agents. On appelle chaîne de valeur, la série de processus qui influencent la valeur des ressources économiques de l'entreprise. La chaîne de valeur REA peut être représentée par un graphe étiqueté et orienté dont chacun des nœuds est un processus d'affaires d'échange ou de conversion. Ces processus ne sont pas décomposables. Les arcs représentent le flux de valeur des ressources entre ces processus.

La chaîne de valeur REA consiste en trois éléments (Hruby, 2006) : (1) le processus d'échange, (2) le processus de conversion, et (3) le flux de valeur de ressource. Ce dernier est la relation entre les processus REA. Cette relation représente l'entrée et la sortie d'un processus d'affaires. L'orientation de l'arc indique que le processus, au début, ajoute de la valeur à la ressource, et que le processus, à la fin, réduit sa valeur. Chaque flux de valeur doit avoir dans chacune de ses extrémités un processus. Chaque processus doit avoir au moins un flux entrant et un autre sortant. Voici le processus de création de la chaîne de valeur au niveau de l'entreprise (Hruby, 2006) :

1. Considérer le contexte de l'entreprise pour créer un schéma d'échange global de ressource économique. Le diagramme résultant montre le contexte d'affaires de l'entreprise. Ce diagramme est appelé la vue du système de valeur (*Value System View*) par Dunn, Cherrington et Hollander (2004). La Figure 4.14 montre un exemple de contexte d'affaires d'une entreprise qui produit et vend un produit *P*.

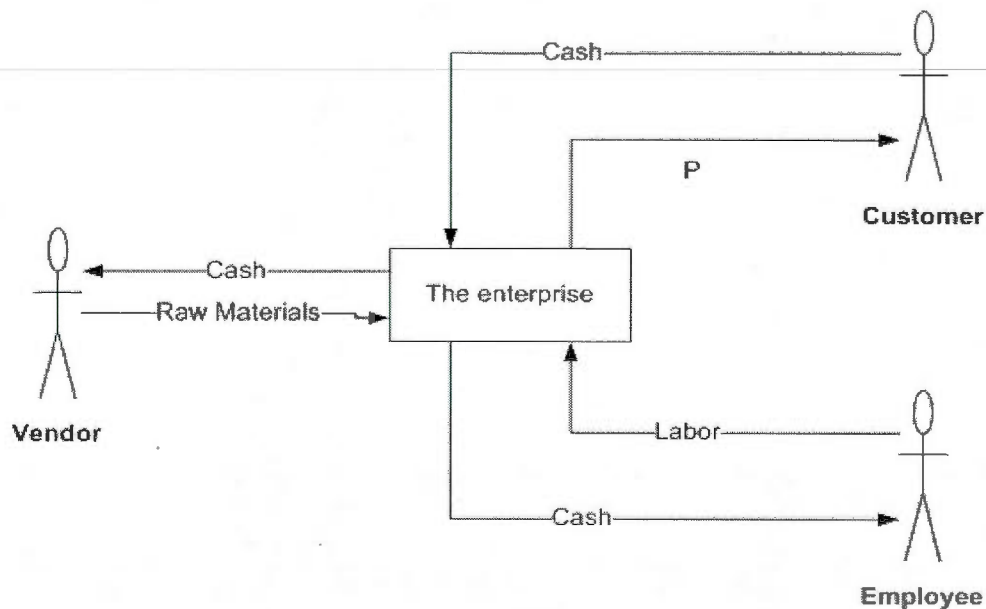


Figure 4.14 Exemple de contexte d'affaires d'une entreprise. (Source : Hruby, 2006)

2. Identifier les processus d'affaires de l'entreprise et les relier par le flux de ressources. Par exemple, à partir du contexte d'affaires illustré dans la Figure 4.14, nous pouvons faire ressortir les processus d'échange suivants : (i) le processus de vente du produit *P*, (ii) le processus d'achat des matières premières, et (iii) le processus d'acquisition de main-d'œuvre qui consiste en un échange de la ressource économique *Labor* pour du *Cash*. L'entreprise a un ou plusieurs processus de conversion. Par exemple, le processus de production du produit *P* à partir des matières premières et de la main-d'œuvre. La Figure 4.15 montre un exemple de chaîne de valeur REA.

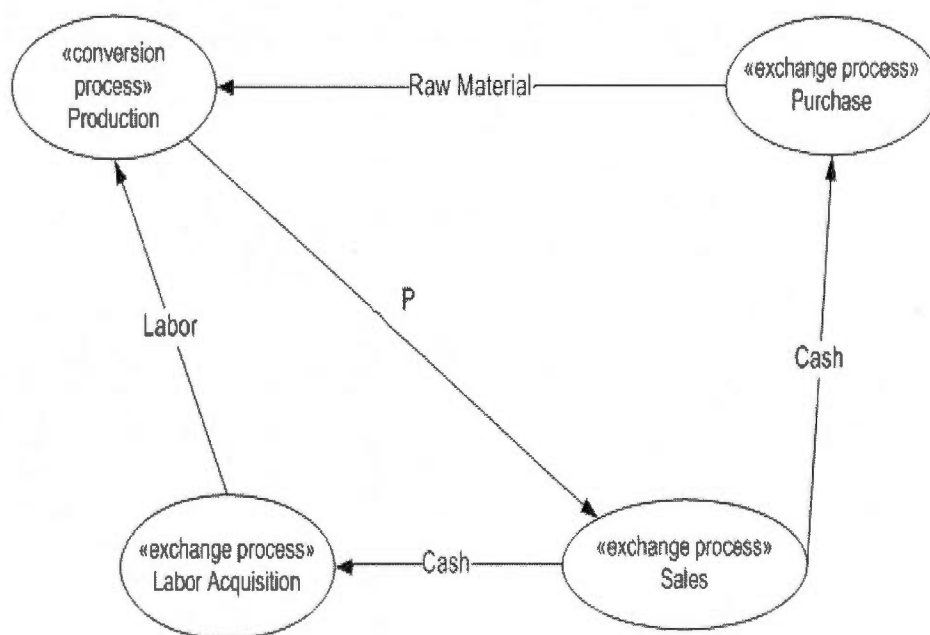


Figure 4.15 La chaîne de valeur initiale. (Source : Hruby, 2006)

3. Décomposer de manière hiérarchique les processus de l'étape 2 pour identifier toutes les ressources que l'entreprise veut planifier, surveiller et contrôler. Le niveau de détail de la décomposition dépend des besoins d'affaires de l'entreprise. Le processus s'arrête au niveau où on ne veut plus planifier, surveiller et contrôler des ressources de l'entreprise.
4. Identifier et ajouter à la chaîne de valeur les processus d'affaires de support tels que les processus de gestion de ressources humaines, de planification, du marketing, et de comptabilité. La Figure 4.16 montre un exemple de chaîne de valeur REA. On notera le processus de planification.

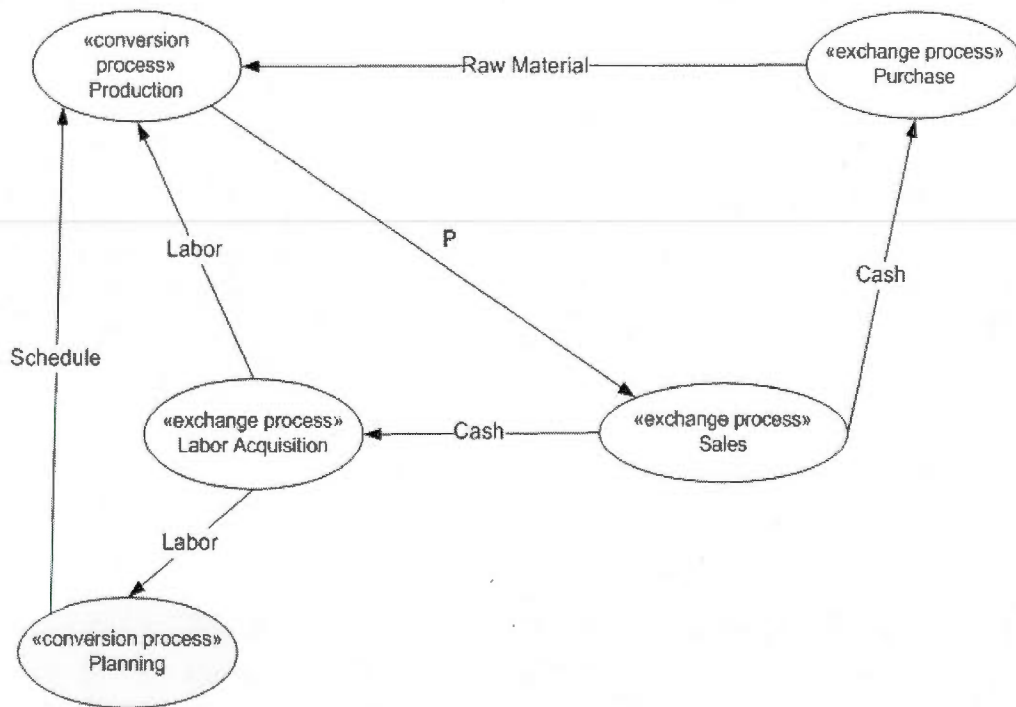


Figure 4.16 La chaîne de valeur finale. (Source : Hruby, 2006)

5. Finalement, il faut vérifier si le diagramme résultant ne viole pas une des règles de REA. Par exemple, il faut s'assurer que chaque ressource économique est reliée à un événement d'incrément et à un événement de décrétement.

La chaîne de valeur REA permet aux analystes et concepteurs de processus de se concentrer sur les processus d'affaires qui ajoutent de la valeur aux produits et services finaux. L'inconvénient de la chaîne de valeur est qu'elle ne permet pas au lecteur du diagramme de savoir par où commencer. Toutefois, certaines heuristiques permettent d'identifier le processus final. Ce dernier, consiste en général en un processus d'échange de type vente ou équivalent (Hruby, 2006).

4.3 Extension du méta-modèle de REA

Depuis sa création, l'ontologie REA a subi des extensions pour répondre à de nouveaux besoins d'affaires. Ces extensions ont été proposées par les auteurs de REA, notamment Geerts and McCarthy, et des acteurs dans le domaine du commerce électronique qui ont intégré REA dans leurs normes (e.g. ISO Open-EDI Business Transaction Ontology, ebXML et UMM). Parmi les extensions majeures de REA, notons les concepts d'engagement, de contrat et de typification.

4.3.1 Le concept d'engagement

Le méta- modèle REA de base ne traite pas le concept d'échanges non instantanés. Dans le domaine d'affaires, les échanges sont généralement négociés d'avance. Le modèle de base ne supporte pas les événements économiques futurs. Nous appelons les promesses d'échange ou de transformation, des engagements (*Commitments*). Ijiri (1975) définit les engagements comme suit: "Agreement to execute an economic event in a well-defined future that will result in either an increase of resources or a decrease of resources."

Dans le but d'étendre ce cadre ontologique, Geerts et McCarthy (2000) proposent l'ajout du concept d'engagement. La Figure 4.17 montre la relation entre l'engagement et l'événement économique.

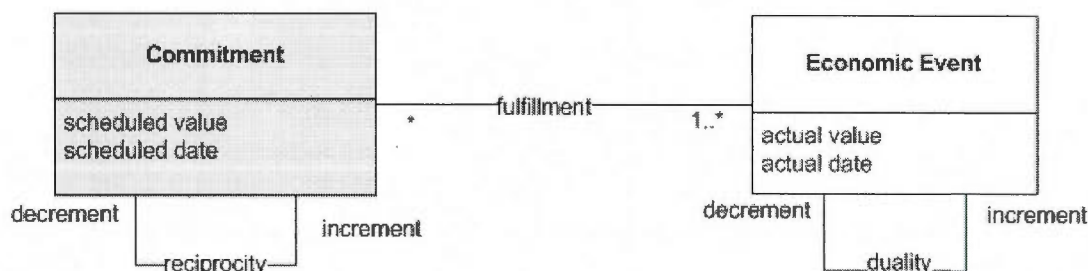


Figure 4.17 L'engagement et l'événement économique en REA. (Source : Hruby, 2006)

Pour chaque processus d'échange ou de conversion, il existe au moins deux engagements : (i) un engagement d'incrément (*Increment Commitment*) qui doit être accompli (relation *fulfillment*) par l'événement économique d'incrément. Cet engagement doit augmenter la valeur de la ressource économique, et (ii) un engagement de décrétement (*Decrement Commitment*) qui doit être accompli par l'événement économique de décrétement. Cet engagement doit diminuer la valeur de la ressource économique. Un engagement peut être rempli par plusieurs événements économiques. Par exemple, un engagement d'achat peut être complété après plusieurs événements de paiement. Un événement économique peut accomplir plusieurs engagements. Par exemple un seul événement de paiement peut compléter plusieurs engagements d'achats.

Les engagements d'incrément sont reliés aux engagements de décrétement par la relation de réciprocité (*reciprocity*). Cette relation porte la même sémantique que la relation de dualité entre les événements économiques d'incrément et de décrétement. Dans un processus d'échange, la relation de réciprocité permet de modéliser les ressources promises pour un échange. Dans un processus de conversion, cette relation permet de modéliser quelle ressource a été réservée en vue d'être utilisée ou consommée pour produire ou transformer une autre ressource. La relation *fulfillment* (ou encore *execution* dans la version originale) entre l'événement économique et l'engagement permet de valider si les engagements ont été exécutés par les événements économiques associés.

4.3.2 Le concept de contrat

Dans sa version de base, l'ontologie REA ne modélisait pas le concept de contrat (*Contract*). Un contrat permet de réglementer le comportement entre les organisations en définissant les promesses d'échange ou de conversion (*Commitment*) ainsi que les clauses d'annulation ou de violation (*Term*). L'ordre d'achat, de vente ou de distribution sont des exemples de contrats REA.

Geerts et McCarthy (2000) ont proposé l'ajout du concept de contrat. Pour une conversion, le contrat représente une cédule (*schedule*) (e.g. cédule de production ou une cédule d'inspection). La Figure 4.18 montre la relation entre le contrat, les engagements, et les termes (Hruby, 2006).

Les engagements représentent les clauses du contrat. Ils représentent le cas idéal de l'échange ou de la conversion économique (Hruby, 2006). Les termes du contrat (*Term*) sont des obligations à exécuter sous certaines conditions. Par exemple, les pénalités à appliquer en cas d'annulation ou de violation des engagements. Nous présentons le patron du contrat REA dans le chapitre 6.

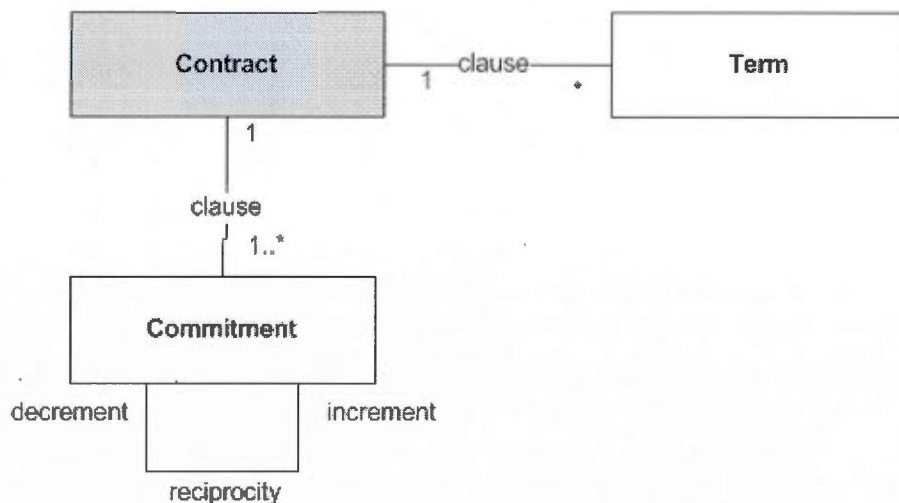


Figure 4.18 Le contrat, l'engagement et le terme. (Source : Hruby, 2006)

4.3.3 La typification

À la base, REA se concentre sur les processus (monde réel), et a une représentation minimale (pauvre) des concepts informationnels (monde pseudo-réel). Geerts et McCarthy (2000) ont enrichi l'ontologie REA par l'ajout du concept de typification¹ qui consiste à aller d'une classe à la méta-classe ou la classe qui la décrit. Par exemple, on peut définir *la compétence* comme type d'agent où chaque *compétence* s'applique à des employés (Geerts et McCarthy, 2000).

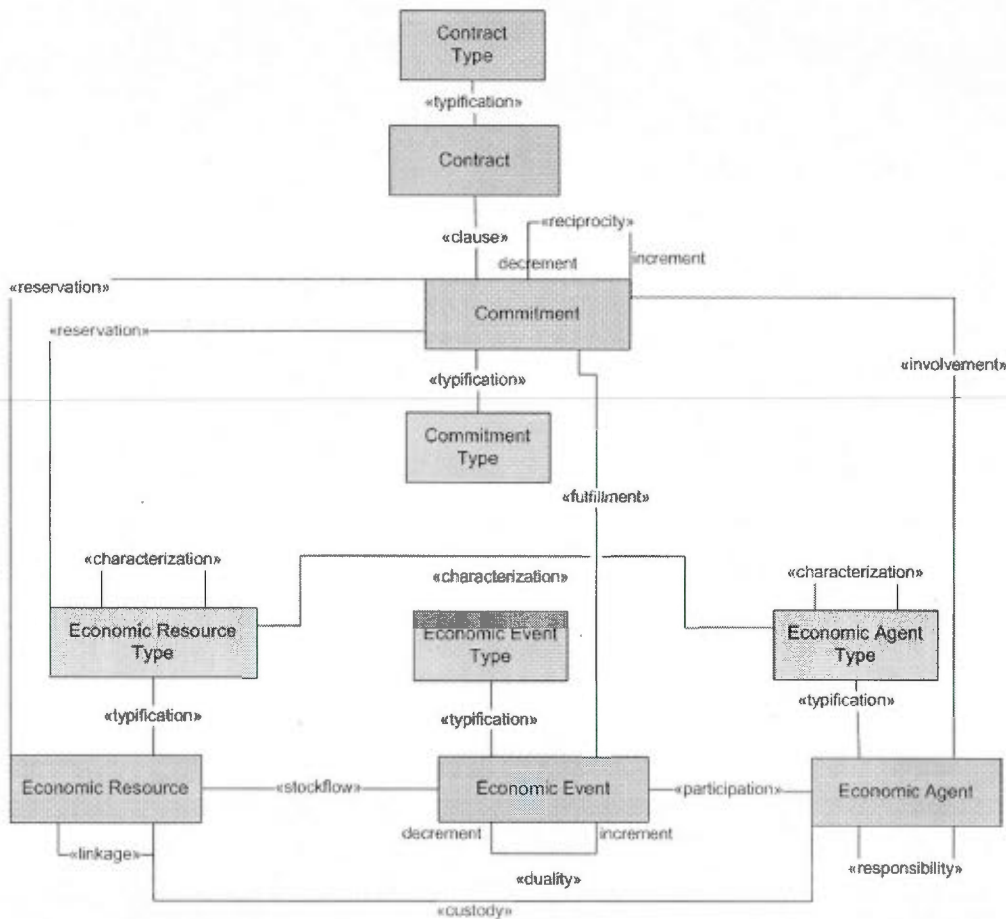


Figure 4.19 L'ontologie étendue de REA. (Source : Hruby, 2006)

¹La typification permet de construire l'infrastructure de connaissance de l'ontologie REA (*The Knowledge Infrastructure*) (Geerts et McCarthy, 2000).

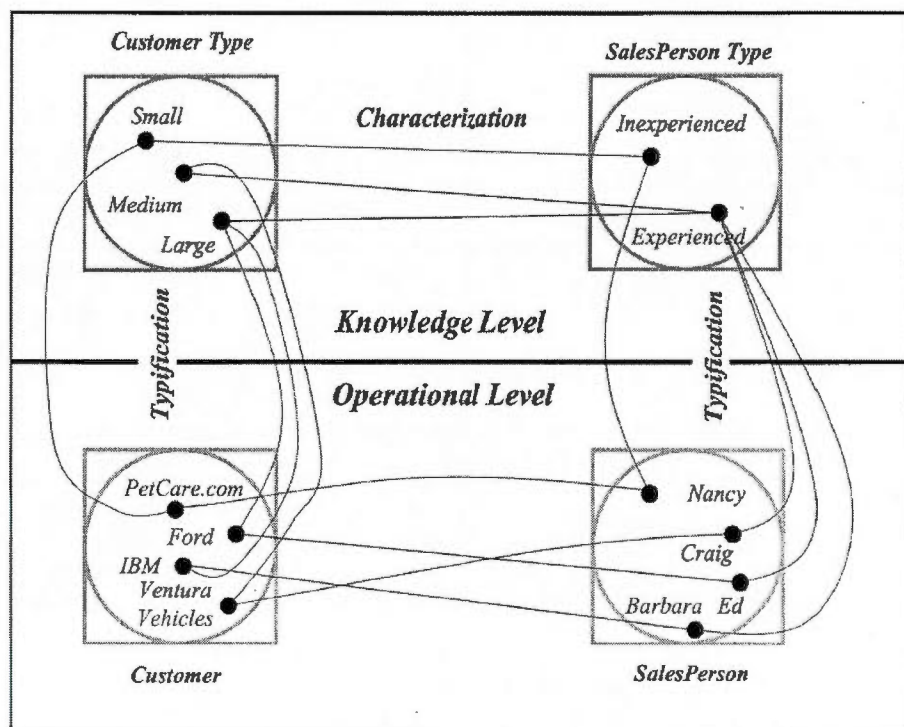


Figure 4.20 Exemple de relation de caractérisation. (Source : Geerts et McCarthy, 2002)

La typification ou encore *spécification* selon Jaquet (2006) et Hruby (2006) est une abstraction qui permet de capturer des informations génériques qui s'appliquent à un groupe de phénomènes. La ressource économique, l'événement économique, l'agent économique, l'obligation, et le contrat sont reliés à leurs types via la relation de *typification*. La Figure 4.19 montre le modèle REA obtenu après l'ajout des concepts d'engagement, de contrat et de typification.

La relation de *caractérisation* (*characterization*) est une relation informative entre différents types. Elle peut être utilisée pour représenter les politiques au sein de l'organisation au niveau de l'infrastructure de connaissances (i.e. les types). Par exemple, une entreprise peut définir la politique suivante: les vendeurs inexpérimentés doivent être affectés aux petites entreprises, alors que les vendeurs expérimentés sont affectés aux moyennes et grandes entreprises (Geerts et McCarthy, 2002). La Figure 4.20 illustre la relation de *caractérisation*.

4.3.4 Les extensions des événements REA

Pour les organisations simples, le modèle REA de base suffit pour décrire leurs processus d'affaires. Cependant, de nos jours, les processus d'affaires deviennent de plus en plus complexes. En effet, plusieurs activités sont souvent impliquées dans un échange, autres que le classique Donner/Recevoir (i.e. *give/take*). Ainsi, plusieurs travaux ont cherché à augmenter la théorie de base de la méthodologie REA pour tenir compte de ces activités qui ne changent pas nécessairement la valeur de la ressource économique.

Dans son approche, David (1997) a introduit de nouveaux types d'événements REA, en plus des événements économiques. En effet, David propose l'ajout *des événements d'affaires* et *des événements informationnels*. Un événement d'affaires est tout événement qui permet à l'organisation de mieux planifier, surveiller et contrôler les événements économiques. Par exemple, créer un devis ou un bon de commande sont des événements d'affaires. Ce type d'événement provoque des changements dans le monde physique et permet de fournir de nouvelles informations qui peuvent être utilisées par l'organisation pour prendre des décisions. Cette définition comprend l'ensemble des événements économiques déjà évoqués. Cependant, ces derniers devraient continuer d'être désignés comme des événements économiques parce que ce terme est plus précis. La relation entre les événements économiques et d'affaires est qu'un événement économique peut être supporté par plusieurs événements d'affaires. Par exemple, la création du bon de commande dans un processus d'approvisionnement est un événement d'affaires car il ne change pas la valeur de la ressource, mais il fournit à l'organisation l'information suivante : l'achat des produits est planifié et les prix sont négociés.

L'événement informationnel est l'activité du processus qui sert à capturer, manipuler ou à communiquer des informations sur les ressources, les agents et les événements économiques et d'affaires. Il s'agit des activités de manipulation de données seulement. Par exemple, générer des factures, imprimer des rapports et afficher l'historique d'un client (David, 1997).

Dunn, Cherrington et Hollander (2004) présentent une autre classification des événements REA en raffinant les événements d'affaires en événements opérationnels, de

décision et de gestion. Pour les besoins de notre approche, nous ne considérons pas cette classification.

4.4 Conclusion

Nous avons présenté dans ce chapitre des concepts avancés de l'ontologie d'affaires REA. Nous avons commencé par présenter notre motivation quant à l'utilisation d'une ontologie d'affaires pour notre approche de réutilisation et d'adaptation de processus d'affaires. En effet, une telle ontologie permet de modéliser nos processus d'affaires, de développer notre catalogue de questions génériques, et nous permettra de mettre en œuvre des spécialisations génériques qui ont un sens métier. Nous avons, par la suite, montré comment modéliser un processus d'affaires avec REA. Pour cela, nous avons présenté les deux patrons de processus REA : (i) le patron d'échange, qui modélise le transfert de ressources économiques entre les agents, et (ii) le patron de conversion, qui modélise la création de nouvelles ressources économiques ou la transformation de ressources économiques existantes par l'utilisation ou la consommation d'autres ressources.

Nous avons ensuite présenté quelques extensions ontologiques de REA. Ces extensions ont été nécessaires pour répondre aux besoins d'affaires grandissant des processus de nos jours. Enfin, nous avons présenté de nouveaux types d'événements REA. Particulièrement, les événements d'affaires et les événements informationnels tels que proposés par David (1997). Dans les chapitres suivants, nous montrerons, avec plus de détails, notre approche de spécification de processus d'affaires en utilisant les concepts REA présentés dans ce chapitre.

CHAPITRE V

REPRÉSENTATION ET CLASSIFICATION DES PROCESSUS D'AFFAIRES

Dans ce travail, nous proposons une nouvelle approche de réutilisation et d'adaptation d'un catalogue de processus génériques. Nous avons présenté une première démarche de spécification de processus d'affaires dite *métalangage* (chap. 3). Nous avons montré que cette dernière présentait quelques problèmes. Nous avons par la suite proposé une nouvelle solution plus prometteuse basée sur l'ontologie d'affaires REA. Nous avons présenté un survol de cette nouvelle approche avec un exemple dans le chapitre 3 (sect. 3.3). Finalement, nous avons décrit les concepts de l'ontologie REA dans le chapitre 4.

Ce chapitre est dédié à l'aspect de classification et de représentation de processus d'affaires au sein d'un catalogue commun. La section 5.1 décrit notre méthodologie de modélisation de processus générique. Nous illustrerons notre approche de représentation par un exemple. Nous présentons, à la section 5.2, notre implémentation du méta-modèle et des vues de processus d'affaires. Finalement, la section 5.3 discute de notre organisation du catalogue de processus génériques.

5.1 Problématique de la représentation de processus génériques

La modélisation de processus d'affaires permet de comprendre les besoins des entreprises et le contexte des systèmes d'information qui leurs sont associés (Phalp et Shepperd, 2000). Selon Ould (1995), la modélisation de processus d'affaires permet de les décrire, de les analyser et de

les mettre en œuvre. Nous croyons qu'une représentation précise des processus génériques nous permettra de les réutiliser et de les adapter pour les besoins spécifiques des organisations.

5.1.1 Qu'est-ce qu'un processus d'affaires générique ?

Jusqu'ici, nous n'avons pas donné une définition formelle d'un processus générique. Selon UN/CEFACT (2005), un processus d'affaires générique, ou encore processus d'affaires commun (*Common Business Process*) est un processus d'affaires commun à un domaine d'affaires. Ce type de processus doit être indépendant du secteur d'industrie spécifique des organisations (e.g. industrie pharmaceutique). D'après Ciuksys et Caplinskas (2006), un processus d'affaires générique est une abstraction utilisée pour décrire une famille de processus¹ pour chaque fonctionnalité d'affaires. Ce dernier servira comme référence à toutes ses variantes. En d'autres termes, c'est la racine d'une famille de processus. Voici les stratégies les plus connues pour la conception d'un processus d'affaires générique:

1. *Le processus Maximal* : c'est un processus générique obtenu par l'union de toutes les variantes possibles. Pour décrire ce type de processus, Wyner et Jintae (2003) utilisent la sémantique de l'ensemble maximal d'exécution (*Maximal execution set*). Hallerbach, Bauer et Reichert (2010) utilisent, quant à eux, la notion de sur-ensemble (*Superset*). Pour la spécialisation d'un processus d'affaires, seulement l'opération de suppression est appliquée.
2. *Le processus Minimal* : c'est un processus générique obtenu par l'intersection de toutes les variantes possibles (Hallerbach, Bauer et Reichert, 2010). Pour décrire ce type de processus, Wyner et Jintae (2003) utilisent la sémantique de l'ensemble minimale d'exécution (*Minimal execution set*). La spécialisation de ce type de processus permet seulement l'ajout et le déplacement. L'opération de suppression n'est pas permise.
3. *Le processus standard* : c'est le processus standard relativement à un domaine d'affaires. Ce type de processus d'affaires n'existe pas pour tous les domaines

¹ La famille d'un processus d'affaires est la hiérarchie de tous les processus obtenus par la spécification de ce processus.

(Hallerbach, Bauer et Reichert, 2010). À moins que ce processus soit le processus maximal ou le processus minimal, il n'y a pas de restriction au niveau des opérations de spécialisation. On peut ajouter, supprimer et déplacer.

4. *Le processus le plus utilisé* : ce processus correspond à la variante la plus utilisée parmi toutes les variantes ou celles connues dans un domaine d'affaires. Ce type de processus permet de réduire le nombre de transformations (Hallerbach, Bauer et Reichert, 2010). Comme le *processus standard*, il n'y a pas de restriction au niveau des opérations de spécialisation. On peut ajouter, supprimer et déplacer.
5. *Le processus ayant la distance moyenne minimale* : grâce aux techniques de forage structurel (*structural mining*), on peut découvrir le modèle de processus ayant la distance moyenne minimale parmi l'ensemble des variantes. Plusieurs algorithmes, dans le domaine du forage, permettent d'extraire ce type de modèles (Bae *et al.*, 2006 ; Gunther *et al.*, 2006). Li, Reichert et Wombacher (2008) présentent un algorithme de forage basé sur l'analyse quantitative qui permet d'extraire le processus de base avec une complexité polynomiale ($O(n^3)$ où n correspond au nombre de variantes). Ici aussi, on peut ajouter, supprimer et déplacer lors de la spécialisation.

Comme dans Provop (Hallerbach, Bauer et Reichert, 2010), notre approche n'impose pas de stratégie particulière. Le concepteur a le libre choix d'opter pour l'une ou l'autre des stratégies afin d'ajouter un nouveau processus générique dans notre référentiel car notre spécialisation supporte les opérations d'ajout, de suppression et de déplacement. En effet, nous n'imposons aucune restriction au processus d'affaires générique puisque tout processus peut être classifié en un processus d'affaires REA d'échange ou de conversion. La Figure 5.1 montre le processus générique minimal.

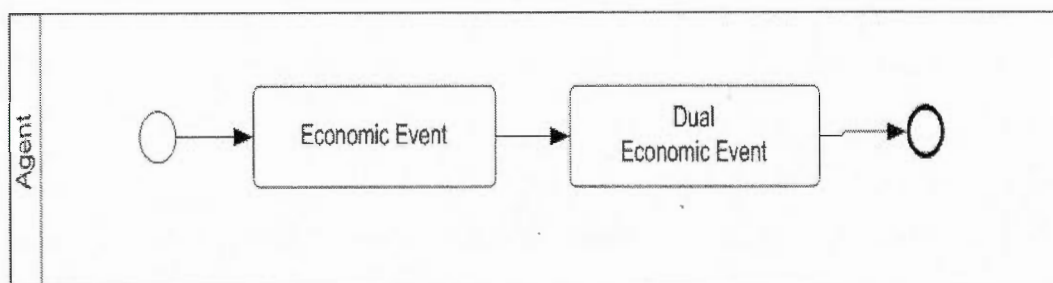


Figure 5.1 Le processus générique minimal.

Ce processus générique peut être facilement étendu par l'ajout de nouvelles étapes. Par exemple, on peut ajouter un événement d'instigation et un événement d'engagement mutuel (contrat ou cédule) tel que proposé par Dunn, Cherrington et Hollander (2004), ou ajouter l'étape de *Post-Actualisation* du modèle ISO Open-EDI (ISO/IEC, 2010). Une telle modification du processus de base n'aura pas d'incidence sur notre approche. En effet, notre spécialisation ne repose pas sur la connaissance préalable de l'ensemble des variations qui doivent être rattachées aux étapes fonctionnelles du processus.

5.1.2 Une vue REA ?

Avant de pouvoir réutiliser un processus d'affaires, il faut d'abord le modéliser et le télécharger dans le référentiel. La modélisation de processus générique doit faire ressortir les deux concepts fondamentaux d'orchestration et de chorégraphie. L'orchestration permet de décrire les successions des différentes étapes du processus, leurs branchements et leurs synchronisations. La chorégraphie permet de décrire les interactions entre plusieurs partenaires d'affaires. Pour faire ressortir ces deux concepts, nous avons choisi la décomposition en perspectives indépendantes et complémentaires proposée par Curtis, Kellner et Over (1992). En effet, ces derniers proposent quatre vues pour la modélisation de processus d'affaires : *la vue fonctionnelle, la vue dynamique ou comportementale, la vue informationnelle et la vue organisationnelle*. Nous avons présenté ces vues dans le chapitre 2. Nous avons utilisé cette décomposition dans notre première approche (i.e. l'approche métalangage). Suite aux problèmes rencontrés dans cette approche, nous avons montré dans les chapitres 3 et 4 que l'approche ontologique basée sur REA présente une meilleure alternative. Dès lors, nous avons

choisi de remplacer la *vue fonctionnelle* par une nouvelle *vue REA*. Cette décision est justifiée par : (1) la vue REA présente les mêmes concepts que la vue fonctionnelle, tout en montrant les agents impliqués dans l'échange ou la conversion, et (2) les vues d'un processus d'affaires sont des instances de notre méta-modèle basé sur l'ontologie REA. Par conséquent, la vue REA est un choix naturel pour représenter les activités à accomplir et les ressources produites et consommées par ces activités.

5.1.3 Comment représenter un processus générique ?

Comme nous l'avons mentionné, notre approche de réutilisation de processus d'affaires n'impose pas de stratégie particulière pour la conception du processus générique. Notre méthodologie de représentation du processus d'affaires générique passe par une étape d'analyse suivie d'une étape de conception. L'étape d'analyse, ou encore de décomposition fonctionnelle, consiste à décomposer le processus d'affaires en entrée en un ensemble de sous processus REA élémentaires. Par la suite, une autre étape d'analyse consiste à classifier l'ensemble des événements économiques, d'affaires et informationnels. L'étape de conception consiste à modéliser chaque processus identifié dans l'étape d'analyse selon le patron REA d'échange ou de conversion. À la fin de cette étape de conception, il faut décomposer chacun de ces processus REA en quatre vues de processus soit, la vue REA, la vue dynamique, la vue organisationnelle et la vue informationnelle. La Figure 5.2 décrit notre processus d'analyse et de conception du processus générique.

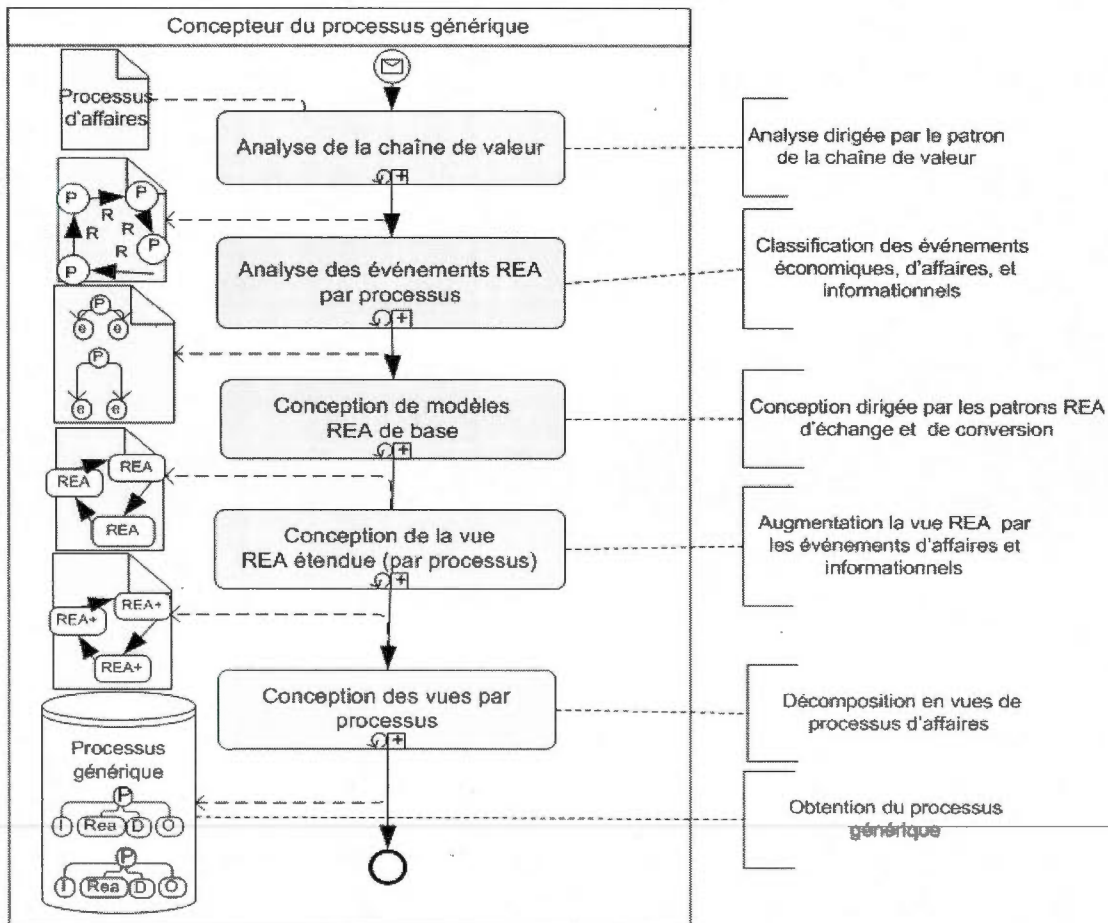


Figure 5.2 Processus de représentation du processus générique.

5.1.4 L'analyse du processus générique

Étant donné un processus d'affaires en entrée, décrit en BPMN par exemple, l'étape d'analyse consiste en deux sous étapes : (1) une décomposition du processus original en une chaîne de valeur de processus REA élémentaires, et (2) l'identification des événements économiques, d'affaires et informationnels.

5.1.4.1 La décomposition du processus original en une chaîne de valeur

La création de la chaîne de valeur REA est un processus de décomposition itératif jusqu'à l'obtention d'une chaîne de processus élémentaires reliés par les ressources. Les processus

élémentaires REA représentent le niveau de granularité le plus bas dans la chaîne. Le but est d'obtenir un graphe étiqueté orienté dont chacun des nœuds est un processus d'affaires de conversion ou d'échange. Les arcs sont étiquetés par les ressources économiques. Cette décomposition suit le patron REA de la chaîne de valeur décrit dans le chapitre 4 (sect. 4.2). La chaîne de valeur REA se compose de trois éléments : (1) le processus d'affaires d'échange, (2) le processus d'affaires de conversion, et (3) le flux de ressources économiques. Ce dernier représente la relation entre les processus REA.

5.1.4.2 La classification des événements REA

La deuxième étape d'analyse consiste à classer les événements en se basant sur l'approche de David (1997), qui propose l'ajout des événements d'affaires et des événements informationnels. Selon David, les événements économiques *seuls* ne permettent pas l'analyse des processus d'affaires de nos jours. En effet, ces nouveaux processus, qui sont de plus en plus complexes, impliquent d'autres événements qui ne reflètent pas nécessairement des changements au niveau des ressources économiques. Les événements d'affaires supportent les événements économiques tout en provoquant des changements dans le monde physique. Ce type d'événements permet de fournir à l'organisation de nouvelles informations qui peuvent être utilisées pour prendre des décisions d'affaires. Par exemple, l'événement *placer une commande d'achat* est un événement d'affaires qui fournit à l'organisation de nouvelles informations : les biens sont cédulés et les prix sont négociés.

Les événements informationnels sont des activités de manipulation de données seulement. Ils servent à capturer, manipuler ou communiquer des informations sur les ressources, les agents et les événements économiques et d'affaires. Par exemple, nous pouvons citer : produire une facture, imprimer un rapport ou consulter les données historiques des transactions. À l'instar de plusieurs spécialistes REA qui ne conseillent pas l'ajout des événements informationnels, nous ne conservons que les événements informationnels qui ont une incidence sur le processus. Plus précisément, nous conservons ceux ayant une incidence sur le déroulement d'un ou plusieurs événements d'affaires ou économiques. Cette incidence se traduit par un changement dans la partie dynamique du processus si l'événement informationnel est supprimé ou déplacé. Prenons l'exemple de gestion des créances. Dans

certain cas, le processus d'échange n'exige pas la matérialisation et l'envoi de la créance (e.g. facture). Cette dernière est calculée et validée en se basant seulement sur le contrat d'échange établi entre les partenaires d'affaires et le bon de réception des ressources économiques. Dans ce cas, l'événement d'affaires qui permet d'établir le paiement dépend de l'existence de l'événement informationnel de matérialisation de la créance. Dès lors, l'absence de l'événement informationnel de matérialisation de créance provoque un changement dans le déroulement du processus d'affaires. Par conséquent, cet événement informationnel sera considéré dans notre approche de spécification de processus d'affaires. Les événements informationnels qui n'ont pas d'incidence sur le déroulement du processus d'affaires ne seront pas considérés dans notre méthodologie.

Nous avons expliqué dans le chapitre 3 que les événements économiques effectués par un partenaire d'affaires ne seront pas modélisés dans la vue REA étendue. En effet, cette stratégie permet d'éliminer les événements en double puisque les événements économiques miroirs (sect. 4.2.3) sont identiques. Par conséquent, nous ne modélisons que les événements économiques effectués par l'organisation.

5.1.5 La conception du processus générique

L'étape d'analyse produit deux artefacts : (1) la chaîne de valeur REA et (2) l'ensemble des événements économiques, d'affaires et informationnels pour chacun des processus de la chaîne de valeur. L'étape de conception consiste à modéliser chaque processus de la chaîne de valeur. Cette conception suivra les deux patrons REA d'échange et de conversion présentés dans le chapitre 4.

Le modèle de base de l'ontologie REA couvre seulement les événements économiques. Il faut ajouter par la suite les événements d'affaires et informationnels à chaque modèle de processus. Le modèle final, obtenu après l'augmentation de la vue REA de base par les événements d'affaires et informationnels, forme *la vue REA étendue*. C'est la vue finale de notre approche. Par la suite, il faut décomposer chacun des processus pour ressortir les vues dynamique, informationnelle et organisationnelle selon la définition de Curtis, Kellner et Over (1992). La Figure 5.3 montre les vues de processus d'affaires.

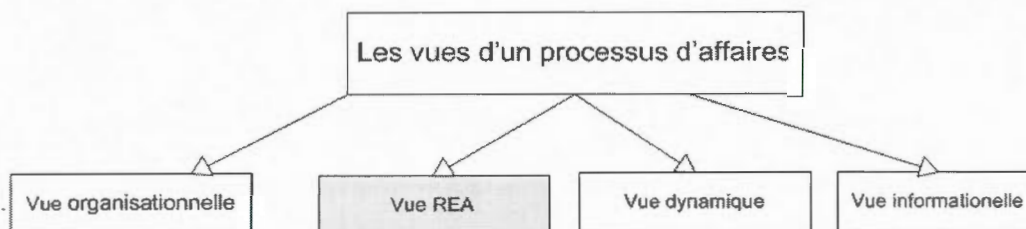


Figure 5.3. Les vues d'un processus d'affaires.

5.1.6 Exemple : représentation du processus d'affaires de vente et distribution

Le processus de vente et distribution consiste en deux processus d'affaires : (i) la vente, et (ii) la distribution. Le processus de vente est un processus d'échange qui consiste à transférer le droit de propriété d'un produit à un acheteur en échange d'une contrepartie. La contrepartie est généralement la remise d'une somme d'argent. La distribution est un processus de conversion qui change une des caractéristiques du produit, soit son emplacement. La Figure 5.4 montre le diagramme BPMN du processus de vente et distribution.

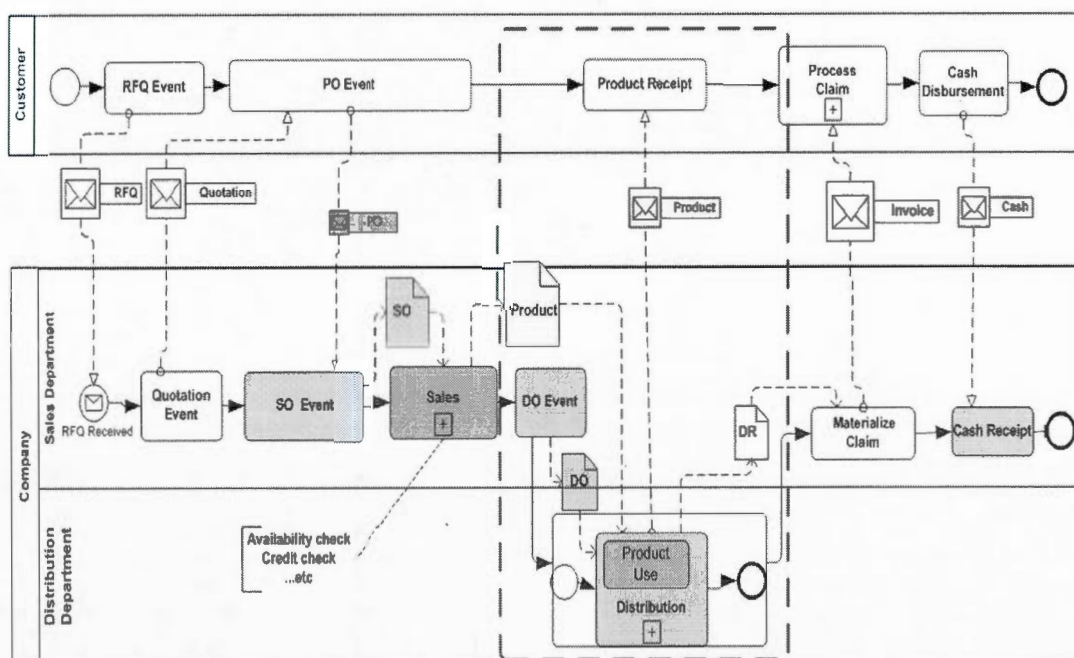


Figure 5.4 Le processus de vente et distribution.

Les ressources RFQ (*Request For Quotation*), Quotation, SO, DO désignent, respectivement, la demande de devis, le devis, le contrat de vente et le contrat de distribution. La structure des ressources RFQ et Quotation est la même que celle du contrat (Hruby, 2006; Murray, 2009). Les agents économiques négocient les obligations futures (*Commitments*) et les termes (*Terms*). Quand les agents parviennent à un accord, les ressources RFQ et Quotation seront transformées en un contrat de vente liant les agents économiques.

5.1.6.1 L'analyse de la chaîne de valeur

Le processus de vente et distribution (S&D) consiste en deux processus REA. La vente est un processus d'échange. Le processus de vente représente l'image miroir du processus d'approvisionnement déjà présenté dans le chapitre 3 (i.e. c'est le processus d'approvisionnement vu selon la perspective REA du partenaire d'affaires). La distribution est un processus de conversion. Dans le processus générique de la Figure 5.4, c'est l'organisation qui est responsable de distribuer ses produits. Il n'y pas de service de distribution fourni par un tiers. Dès lors, la seule ressource en entrée et en sortie dans ce processus est le produit à distribuer. À la sortie, seulement l'attribut emplacement (*location*) du produit sera modifié. L'événement d'incrément Distribution est lié à l'événement de décrément Product Use. Ce dernier indique que la valeur du produit est diminuée pour l'entreprise pendant la distribution de ce dernier. En effet, le produit ne peut être utilisé pendant ce temps et sa valeur est diminuée. De plus, celui-ci peut être endommagé (Hruby, 2006). La Figure 5.5 montre la chaîne de valeur du processus de vente et distribution.

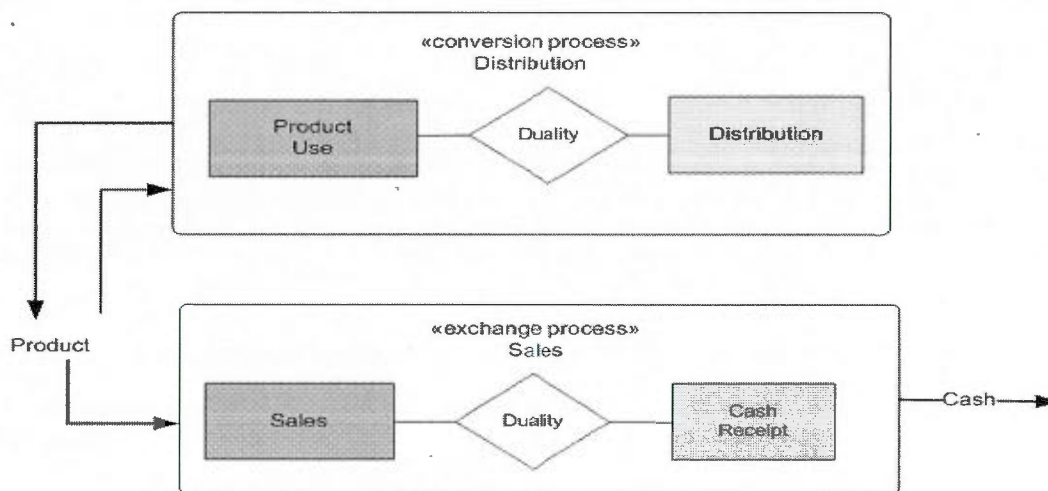


Figure 5.5 La chaîne de valeur du processus de vente et distribution.

5.1.6.2 L'analyse des événements REA

Durant cette étape, l'analyste identifie les différents événements REA en se basant sur l'approche d'analyse de David (1997). Cette analyse doit être effectuée pour chacun des processus de la chaîne de valeur. Les événements économiques sont ceux qui changent la quantité de la ressource économique. À partir du diagramme BPMN (fig. 5.4), nous identifions les événements économiques Sales et Cash Receipt pour le processus de vente, et les événements économiques Distribution et Product Use pour le processus de distribution. Les événements d'affaires permettent à l'organisation de planifier, surveiller et contrôler les événements économiques. Nous identifions les événements d'affaires suivants : RFQ Event, Quotation Event, PO Event, SO Event, DO Event, et Process Claim. Les événements informationnels servent à capturer, manipuler ou communiquer des informations sur les ressources, les agents et les événements économiques et d'affaires. Nous identifions un seul tel événement : Materialize Claim. Cet événement informationnel, qui consiste à matérialiser la créance (e.g. créer une facture) a une incidence sur le processus car l'événement d'affaires Process Claim en dépend. En effet, lorsque la valeur de la créance peut être calculée à partir des engagements, l'organisation n'est pas tenue d'envoyer la

matérialisation de la créance à l'agent partenaire (i.e. le client dans ce cas). Ce dernier peut compléter l'échange en calculant la valeur de la créance automatiquement. Cette méthode se base sur le principe du *Two-way match*. Elle a été développée par la compagnie *General Motors* en 1994. Elle s'appelle aussi *ERS (Evaluated Receipt Settlement)* (Murray, 2009) et se base sur un calcul qui utilise seulement deux composants: le contrat et le bon de réception des ressources économiques. S'il n'y pas de variation, le partenaire d'affaires (le client) envoie le paiement. Quand le processus requiert la matérialisation de créance, l'événement *Process Claim* utilisera la validation classique de type *Three-way match*. Cette dernière se base sur trois composants : la créance, le contrat, et le bon de réception des ressources économiques.

5.1.6.3 Conception des modèles REA de base

La conception de modèles REA se base sur le patron d'échange et de conversion. La Figure 5.6 montre le modèle REA de base du processus de vente.

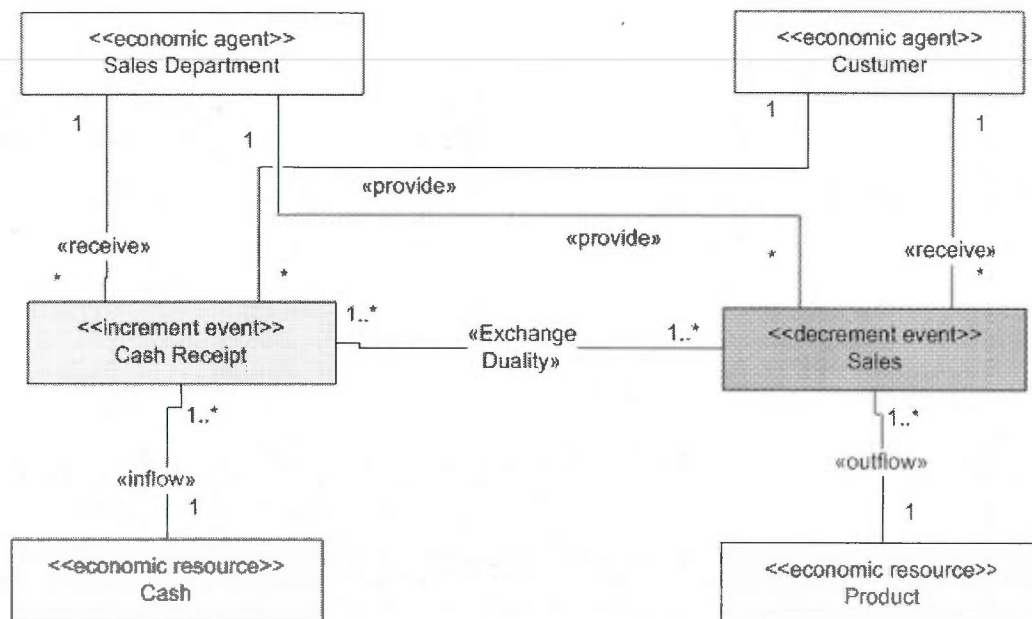


Figure 5.6 Le modèle REA de base du processus de vente.

La Figure 5.7 montre le modèle REA de base du processus de distribution. Ce dernier suit le patron REA de conversion car, au terme de l'événement économique de distribution, l'attribut emplacement (*location*) de la ressource économique est modifié. La Figure 5.8 montre le modèle REA de base de processus de vente et distribution après l'intégration des deux vues ensemble.

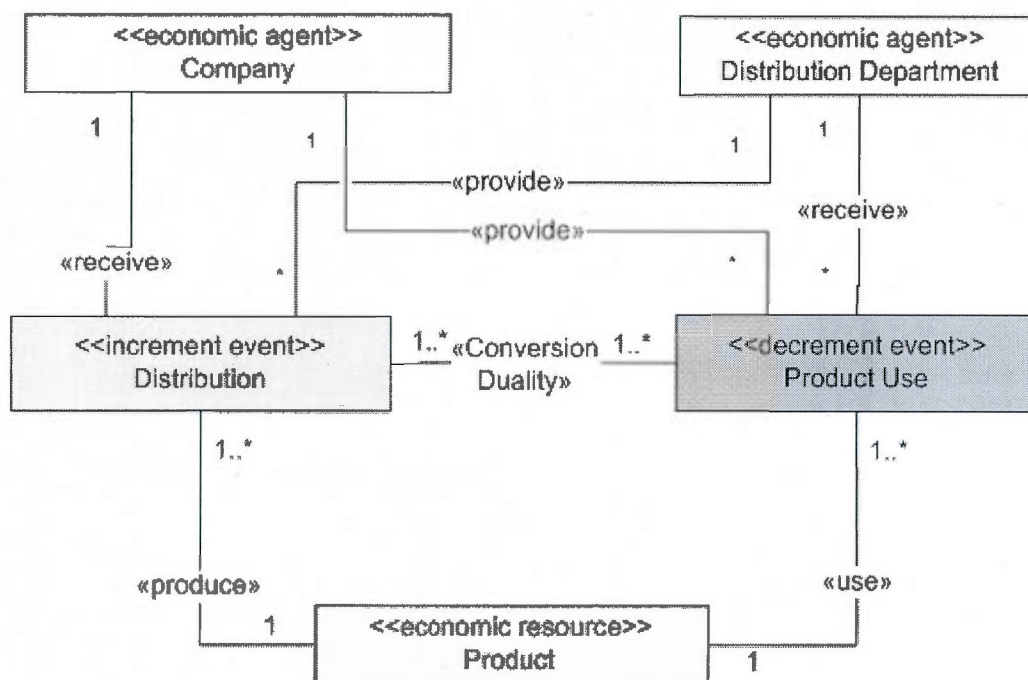


Figure 5.7 Le modèle REA de base du processus de distribution.

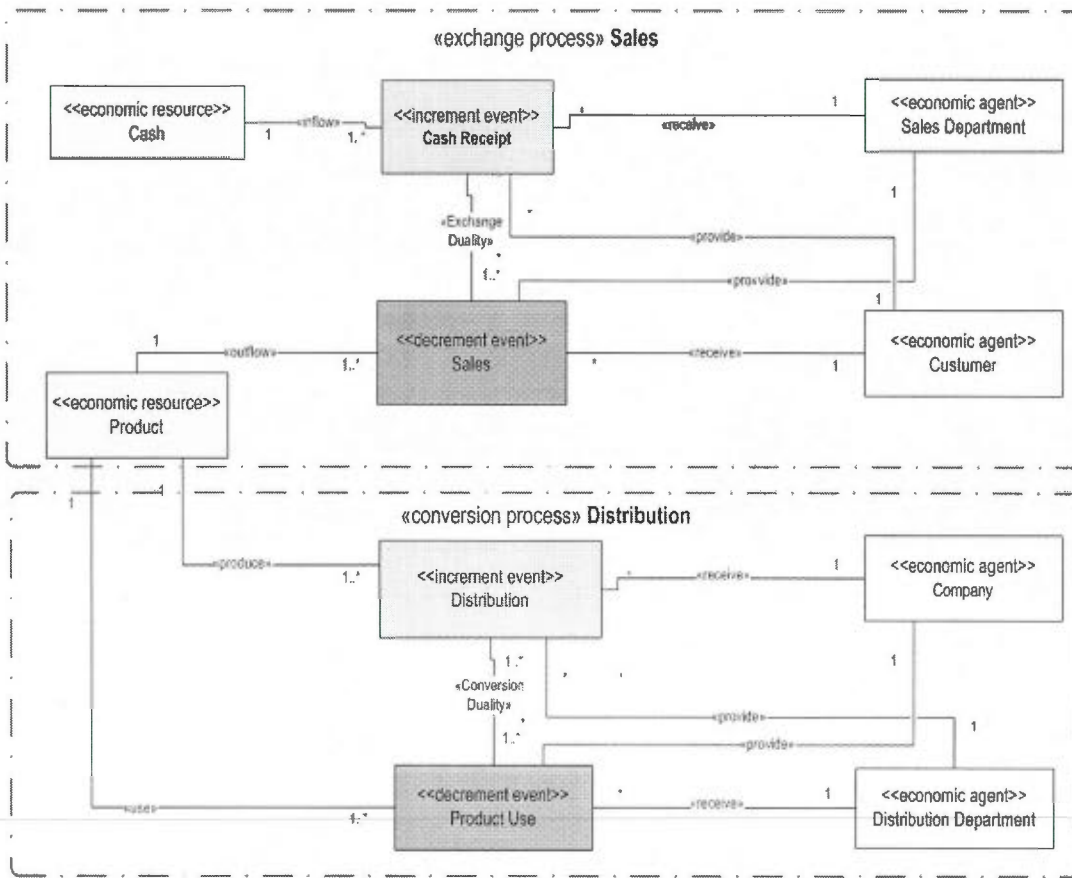


Figure 5.8 Le modèle REA de base du processus de vente et distribution.

5.1.6.4 Modélisation de la vue REA étendue

La conception de la vue REA étendue consiste à augmenter la vue REA de base par les événements d'affaires et informationnels identifiés durant l'étape d'analyse. Nous avons identifié les événements d'affaires RFQ Event, Quotation Event, PO Event, SO Event, et Process Claim. Un seul événement informationnel a été identifié, soit l'événement: Materialize Claim. La Figure 5.9 montre la vue REA du processus de vente en UML.

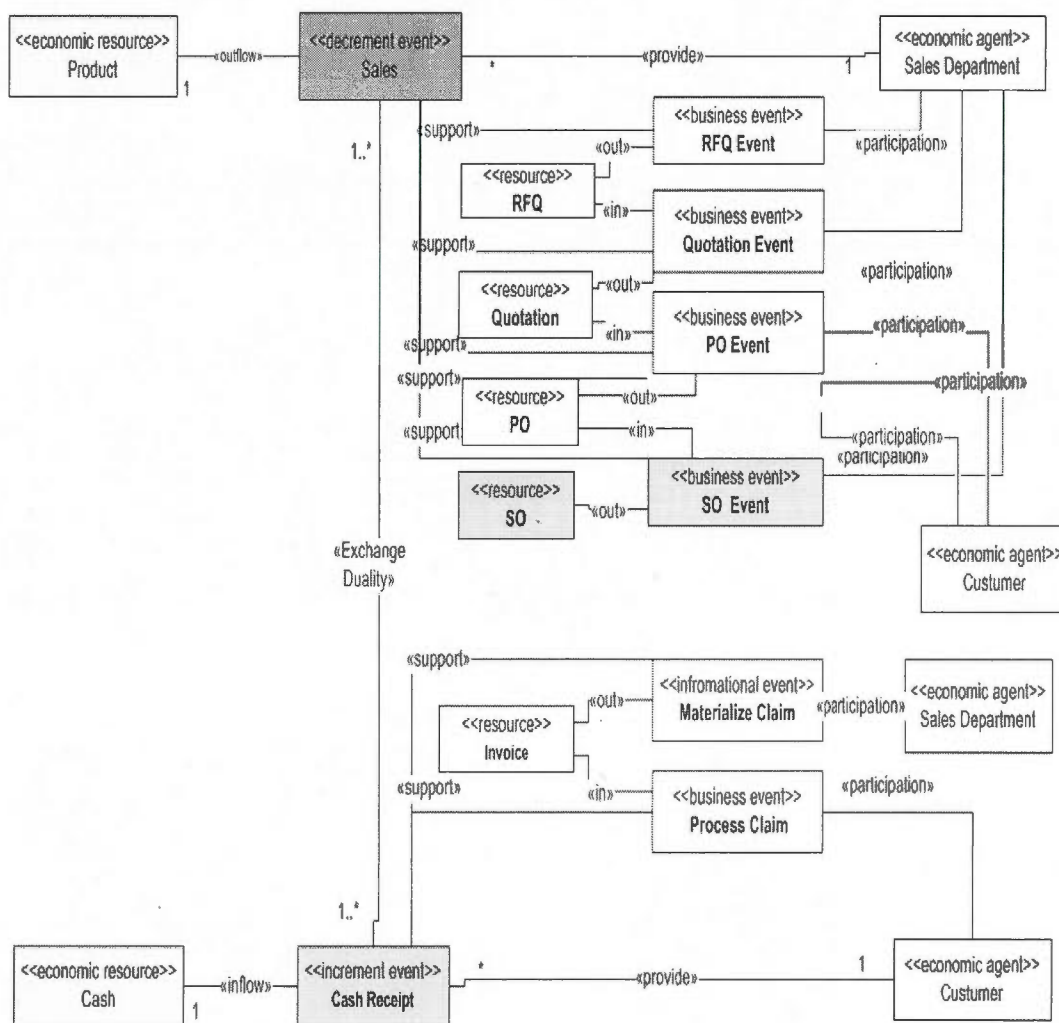


Figure 5.9 La vue REA finale du processus de vente.

Nous appliquons le même processus d'augmentation à tous les processus de la chaîne de valeur. Le diagramme BPMN montre que la vue REA du processus de distribution sera, à son tour, augmentée par l'événement d'affaires de création du contrat de distribution DO Event. La Figure 5.10 montre la vue REA du processus de distribution obtenue après l'ajout de cet événement d'affaires.

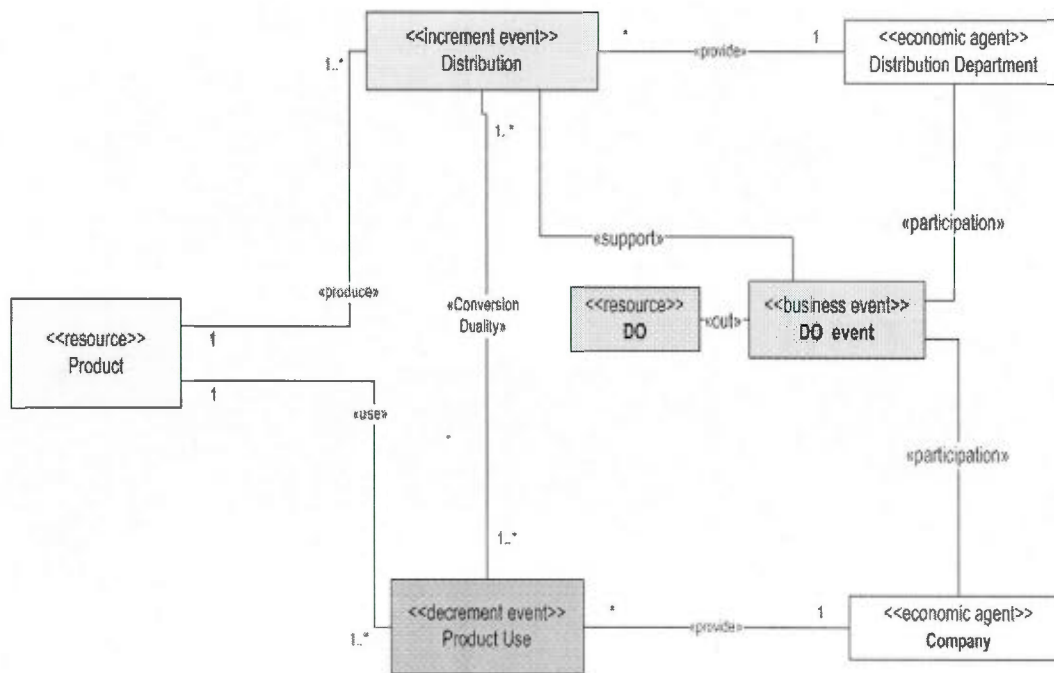


Figure 5.10 La vue REA finale du processus de distribution.

5.1.6.5 Modélisation de la vue organisationnelle et dynamique

La vue dynamique décrit le concept d'orchestration des processus. Elle modélise les séquences d'activités et d'éléments de contrôle qui connectent les différents événements. La vue organisationnelle représente la structure organisationnelle, les rôles et les mécanismes de communication au sein de l'organisation.

Pour illustrer notre approche, nous simplifions la vue organisationnelle. Nous modélisons essentiellement les rôles, les types d'agents ainsi que les fonctions représentées par les événements REA que l'agent exécute. Nous ne montrons pas, par exemple, la structure organisationnelle, ni les mécanismes de communication au sein de l'entreprise. Nous ne pensons pas spécialiser la structure d'une organisation lors de l'application de nos transformations de la vue organisationnelle. Cela ne constitue pas une limite de notre approche. Aussi, tel qu'évoqué au chapitre 3, nous utiliserons le plus possible les concepts graphiques de la norme BPMN pour modéliser la vue dynamique et la vue organisationnelle. Pour faciliter

l'interprétation de ces deux vues et garder le lien avec la description originale du processus d'affaires en BPMN, nous illustrons *les événements économiques miroirs* (sect. 4.2.3). Finalement, les éléments ne faisant pas partie de la vue dynamique ou organisationnelle, mais qui y sont référencés, sont représentés en lignes pointillées. Dans ce cas, la vue représentée (i.e. la vue dynamique ou organisationnelle) fait référence à une autre vue du processus d'affaires.

La Figure 5.11 montre la vue organisationnelle simplifiée de notre processus de vente et distribution. Dans chaque *pool/swimlane*, on montre les responsabilités de l'agent économique correspondant. Les événements illustrés font référence à la vue REA. Les éléments en gras foncé (i.e. les successions, les associations, les événements de début et de fin, etc.) de la Figure 5.12 représentent la vue dynamique.

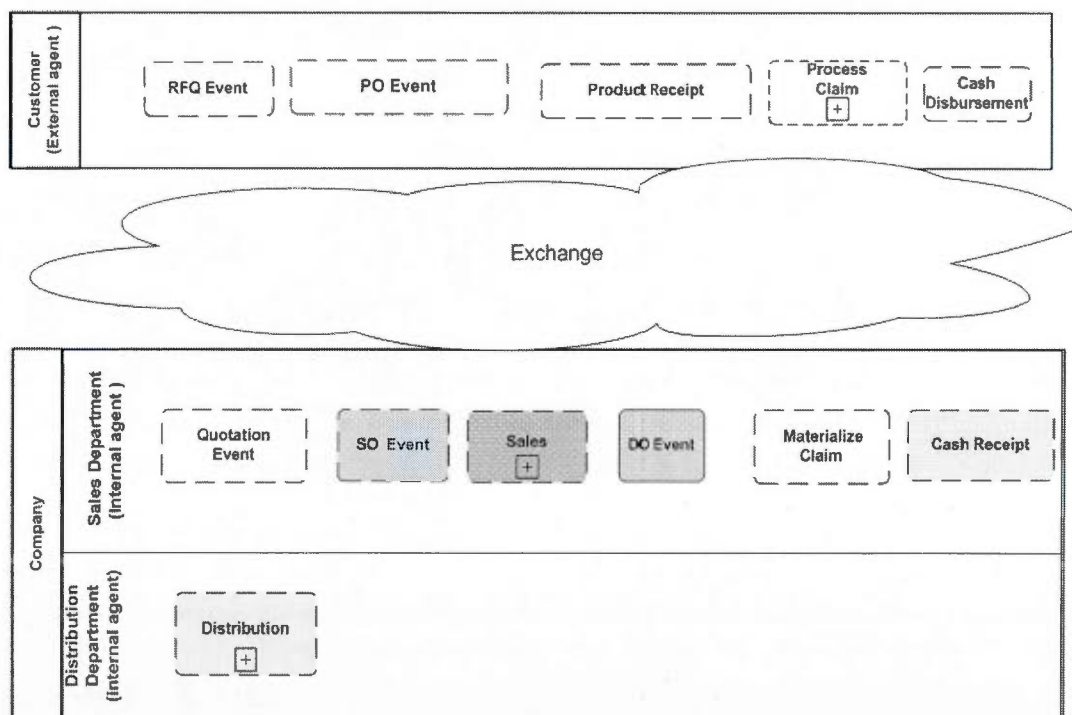


Figure 5.11 La vue organisationnelle du processus de vente et distribution.

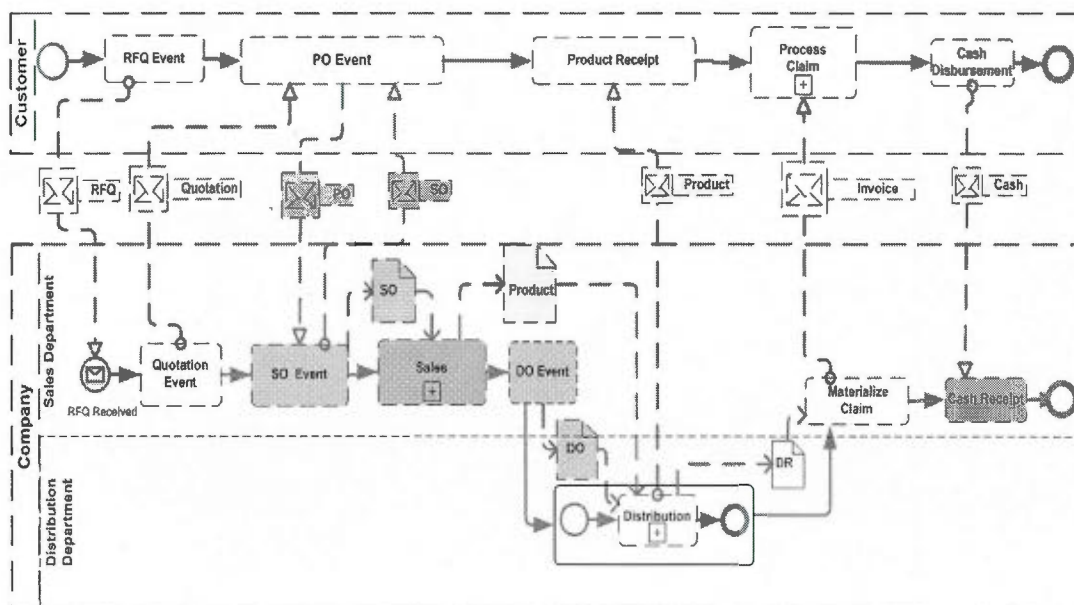


Figure 5.12 La vue dynamique du processus de vente et distribution.

5.1.6.6 Modélisation de la vue informationnelle

La vue informationnelle représente les ressources manipulées par les événements du processus d'affaires. Pour modéliser cette vue, nous utilisons le diagramme de classe du langage UML. Les ressources de notre processus sont RFQ, Quotation, SO, DO, Product, Invoice et Cash. Notre technique de conception de la vue informationnelle est simple. Dans le cas d'existence de contrat (cédule dans le cas de processus de conversion), nous commençons par ce dernier. Un contrat contient les clauses qui sont des obligations d'incrément et des obligations de décrétement. Ces obligations forment les lignes du contrat. Par exemple, le contrat de vente contient des lignes de ventes et des lignes de paiement. Les lignes de vente sont des obligations de décrétement désignant les items à vendre. Les lignes de paiement sont des obligations d'incrément désignant le paiement à recevoir. Les obligations d'incrément et de décrétement sont reliées par la relation de réciprocité. Finalement, les obligations sont reliées aux ressources économiques correspondantes par la relation de réservation. Selon le processus d'affaires, on peut relier une obligation au *type* de la ressource économique. Dans le cas des processus d'échange, nous ajoutons l'entité qui représente la

créance. Dans notre cas, par exemple, la créance REA est matérialisée par une facture qui est envoyée au client. La créance est reliée aux obligations par les relations *materialization* et *settlement*. Quand le processus d'affaires matérialise la créance, nous appliquons le patron d'affaires de matérialisation de créance (*Materialized Claim pattern*) (Hruby, 2006). Enfin, nous ajoutons les ressources RFQ et Quotation. Ces ressources sont produites par des événements d'affaires (voir la vue REA de la Figure 5.9). Celles-ci ont la même structure que le contrat REA et peuvent être reliées aux obligations d'incrément et de décrement. Pour des raisons de simplicité, nous ne représentons les termes du contrat que dans le cas où les processus d'affaires y feraient référence; par exemple, le processus d'affaires de gestion de police d'assurance où l'obligation d'incrément pour le client est instanciée seulement par un terme du contrat. Le diagramme de classe UML de la Figure 5.13 montre la vue informationnelle du processus de vente.

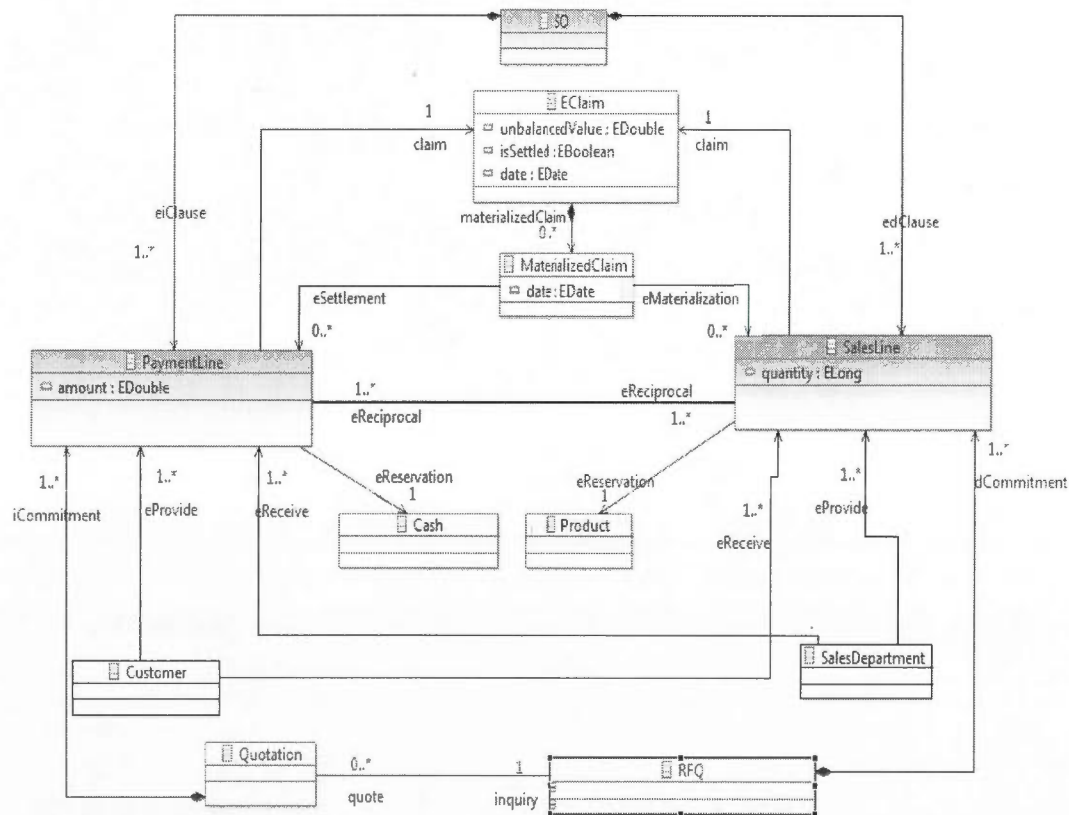


Figure 5.13 La vue informationnelle du processus de vente.

Dans le cas où il n'existerait pas de contrat, la vue informationnelle sera simplifiée par l'élimination du contrat et de ses termes. Les obligations deviennent instantanées et, par conséquent, la créance ne sera plus nécessaire.

Le processus de distribution est un processus de conversion qui change l'emplacement de la ressource économique. Le modèle de la figure 5.14 montre la vue informationnelle du processus de distribution. Notons que le concept de créance n'est pas nécessaire pour les processus de conversion. En effet, la créance économique existe seulement pour les processus d'échange lorsque l'échange n'est pas instantané, créant ainsi un déséquilibre durant le laps de temps qui sépare les événements économiques duaux.

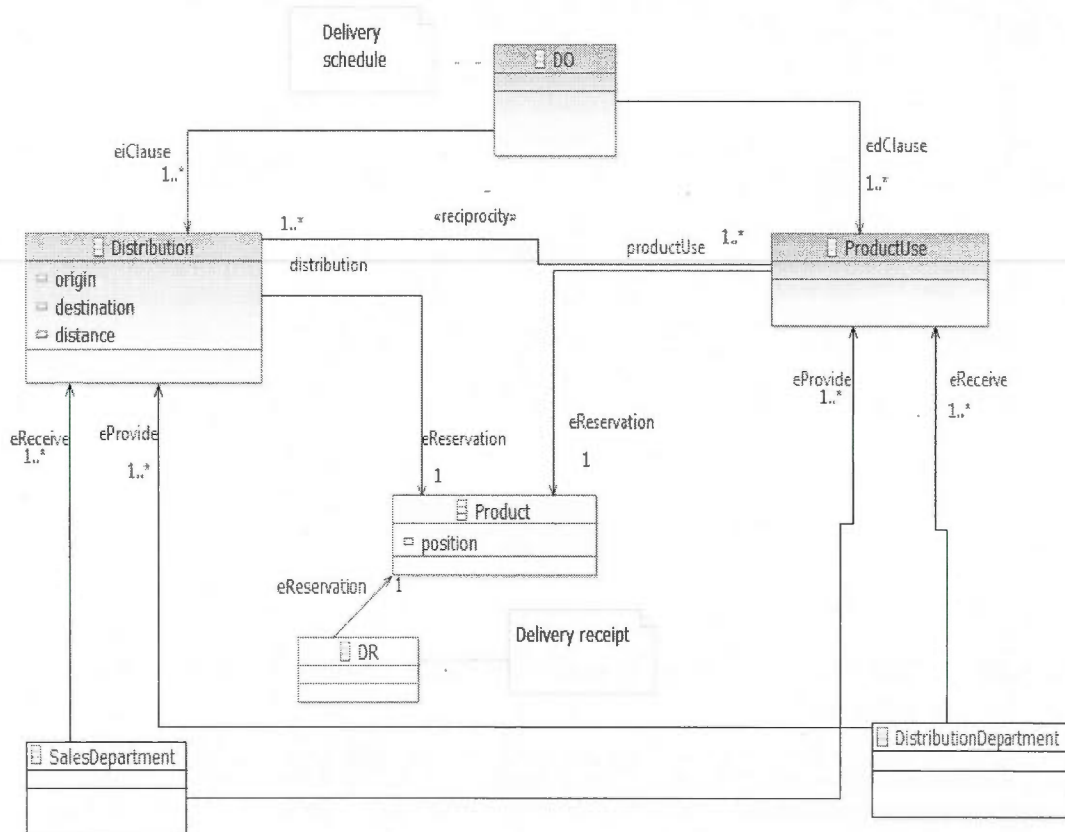


Figure 5.14 La vue informationnelle du processus de distribution.

5.2 Implémentation

Notre méta-modèle se base sur l'ontologie d'affaires REA. À la base, le méta-modèle REA ne supporte pas le concept d'orchestration (sect. 2.1) qui permet l'ordonnement des événements (la vue comportementale). Nous devons alors augmenter ce méta-modèle pour y ajouter le concept d'orchestration. Geerts et McCarthy (2003) ont proposé d'inclure les modèles à base de machines d'états dans l'ontologie REA comme méthode d'ordonnement des événements.

Pour mettre en œuvre l'orchestration, nous avons choisi un langage abstrait indépendant de toute notation graphique et de tout langage d'exécution de processus. Pour ce faire, nous avons considéré les normes et avons le choix entre les deux normes d'échange: XPDL et BPDM. Ce dernier normalise les mécanismes d'échange et représente un méta-modèle commun de collaboration entre entreprises à travers la définition précise de leurs processus d'affaires (OMG, 2008). De plus, le méta-modèle comportemental de BPDM (*BPDM Behavior Model*) est centré sur le concept d'orchestration et intègre une sémantique complète, précise et cohérente de tous les concepts de *workflow* (OMG, 2008). Voici un extrait de la spécification de BPDM (OMG, 2008) :

[...] In BPDM, semantics behind the notations and models are well defined, consistent and sufficient to represent most normal forms of business processes. BPDM is sufficiently precise to model behavioral events (starting, ending, aborting, etc) of processes that allows them to be ordered in time, and have their effects on each other precisely modeled"

Nous avons donc choisi la vue comportementale de BPDM pour notre méta-modèle de processus d'affaires. Notons que la spécification de BPDM inclut une mise en correspondance vers la notation graphique BPMN et vers le langage d'exécution BPEL.

Nous avons mis en œuvre notre méta-modèle avec EMF™ (Eclipse, 2008). Notre outil de modélisation est implémenté sous forme d'un plug-in EMF pour l'environnement Eclipse. La section suivante donne un aperçu du cadre EMF.

5.2.1 Le cadre de modélisation EMF

EMF est un cadre de modélisation développé en Java et destiné à l'environnement Eclipse. EMF offre des fonctionnalités de génération de code Java qui permettent de développer des outils et des applications basés sur des modèles structurés. Cela permet de se concentrer sur le modèle plutôt que sur son implémentation. De plus, il fournit les fondations pour l'interopérabilité entre les différents outils et applications basés sur EMF.

Le cadre de modélisation EMF consiste en trois parties fondamentales : le noyau EMF, EMF.Edit, et EMF.Codegen (Eclipse, 2008).

- 1) Le noyau EMF implémente EMOF (*Essential Meta Object Facility*). Il supporte la notification des changements au niveau des modèles, la persistance et inclut une API réflexive efficace pour manipuler des objets EMF génériques. Le noyau EMF est basé sur le méta-modèle illustré dans l'appendice B. Le modèle utilisé pour représenter les modèles EMF est *Ecore Model*. Les interfaces des éléments de ce modèle sont regroupées dans le paquetage `org.eclipse.emf.ecore`. EMF fournit plusieurs fonctionnalités de sérialisation de modèles. Bien qu'il supporte plusieurs formats, EMF utilise la norme XMI (*XML Metadata Interchange*) (OMG, 2003b) comme forme canonique pour définir les modèles. Il y a plusieurs façons pour créer un modèle EMF, à savoir :

- À partir d'un document XMI;
- À partir d'un schéma XML;
- À partir d'un document XMI exporté depuis un outil de modélisation tel que Rational Rose; ou
- À partir d'interfaces Java dûment annotées avec les propriétés du modèle.

- 2) EMF.Edit est un cadre qui permet de générer des éditeurs pour les modèles EMF. Il offre des classes génériques et réutilisables qui permettent aux modèles d'être affichés en utilisant des éditeurs graphiques (e.g. JFace). EMF.Edit offre des mécanismes génériques de gestion des commandes graphiques tels que la gestion de la commande défaire/refaire (*undo and redo*).

- 3) EMF.Codegen est le générateur d'EMF. Il permet la génération du code à partir des modèles. Plus précisément, il offre trois niveaux de génération : (1) le modèle, (2) les adaptateurs, et (3) l'éditeur EMF. Le code généré dans le niveau modèle inclut les interfaces Java et les classes d'implémentation de toutes les classes du modèle ainsi qu'une classe Fabrique (*factory*) et une classe Paquetage (*Package*). La classe *Fabrique* permet de créer les instances de toute classe non abstraite du modèle. La classe *Paquetage* offre des accesseurs de tous les éléments dans le modèle incluant les classes, les attributs, les types de données et les références. Les adaptateurs sont des classes (*ItemProviders*) qui adaptent les classes du modèle pour les fonctions d'édition et d'affichage. Le générateur d'EMF permet aussi la génération de tests unitaires. EMF.Codegen inclut une interface graphique qui permet de spécifier les options de génération. Il exploite le composant JDT (*Java Development Tooling*) d'Eclipse. Finalement, notons que la mise en œuvre d'Ecore (noyau d'EMF) est générée par EMF.Codegen.

Le méta-modèle EMF est un modèle EMF. La Figure 5.15 montre la hiérarchie du modèle Ecore. Cette hiérarchie a pour racine l'interface `EObject`. Les classes `EModelElement`, `ENamedElement`, `ETypedElement`, `EClassifier`, et `EStructuralFeature` sont des classes abstraites. `EClass` représente les classes EMF. Une classe Ecore est identifiée par un nom et peut contenir des attributs (`EAttribute`), des opérations (`EOperation`) et des références (`EReference`).

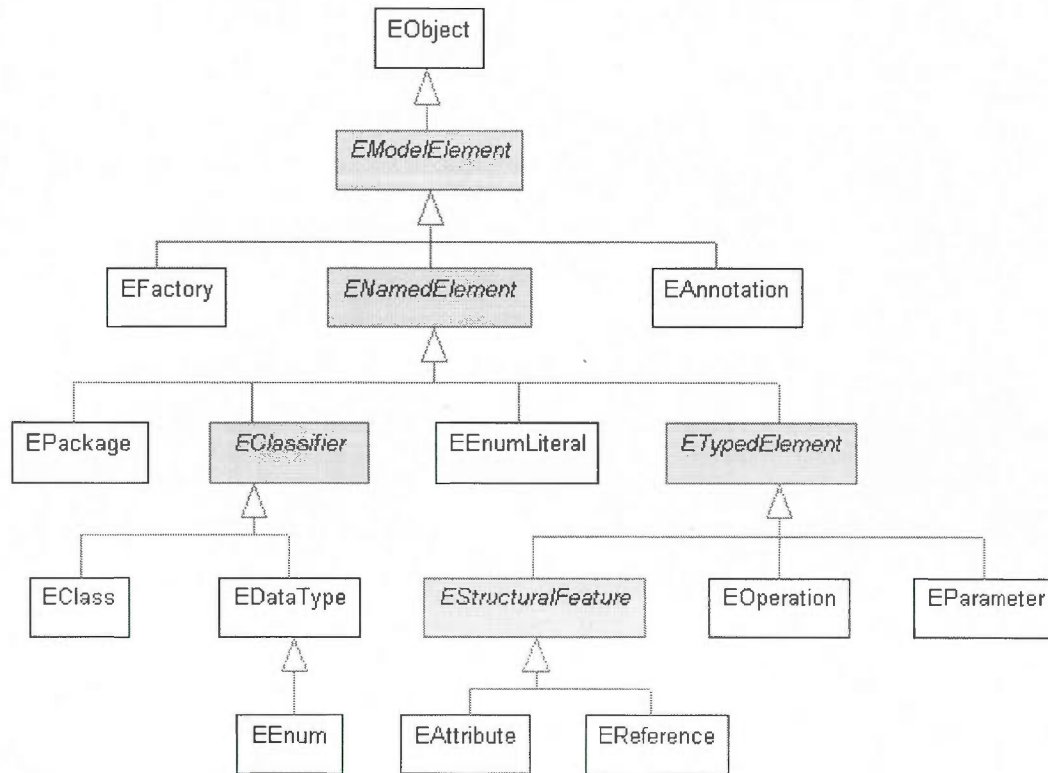


Figure 5.15 Hiérarchie des classes du package Ecore du cadre EMF. (Source : Eclipse, 2008)

Le méta-modèle EMF est présenté dans l'appendice B. Dans ce modèle, un attribut est une instance de l'élément `EAttribute`. Comme la méta-classe `AssociationEnd` dans UML, une instance de `EReference` représente une extrémité d'association entre deux classes. Une référence, tout comme un attribut, a un nom et un type. Le type d'une référence est l'instance de la classe qui est à l'autre extrémité de l'association. Si l'association est navigable dans la direction opposée, il y aura une deuxième référence (`EOpposite`). Le cas échéant, une association est définie par deux instances opposées de `EReference`.

5.2.2 Le méta-modèle REA de base

Comme nous l'avons mentionné, notre méta-modèle de processus d'affaires se base sur l'ontologie d'affaires REA. Nous avons inclus les patrons d'affaires *commitment*,

contract/schedule, linkage, typification, responsibility, custody et policy (Hruby, 2006). Plusieurs de ces patrons font partie de l'ontologie REA étendue.

Pour distinguer les agents internes des agents externes, nous utilisons les concepts BPDM de *RolePerformer* et *InteractionRole*. Un agent interne est représenté par le concept de *RolePerformer*. Un agent externe à l'organisation est représenté par le concept *InteractionRole*. Nous rappelons qu'un agent externe au sens de REA peut être un agent interne à l'organisation. La Figure 5.16 montre notre mise en œuvre de l'ontologie d'affaires REA avec EMF.

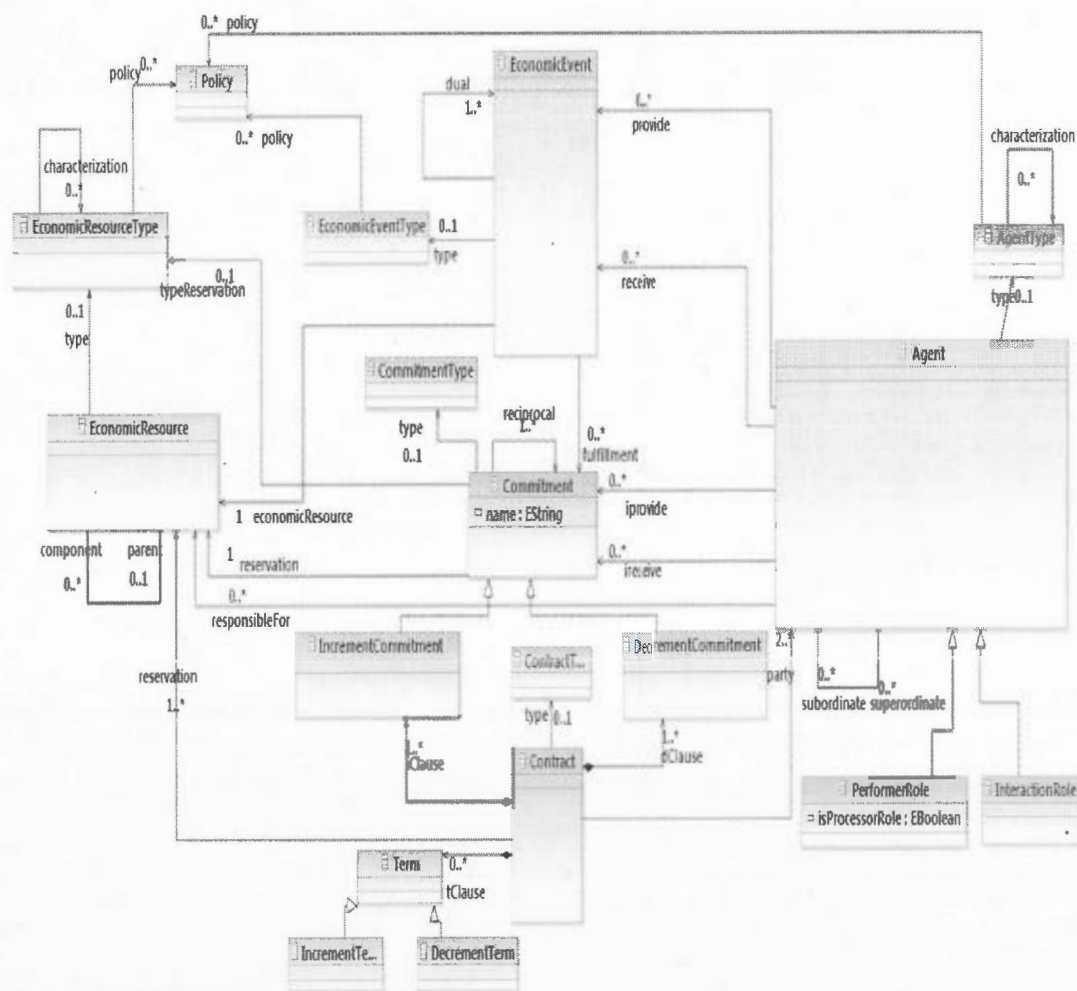


Figure 5.16 La mise en œuvre de l'ontologie REA.

5.2.3 Le méta-modèle REA étendu

La Figure 5.17 montre l'essentiel de notre méta-modèle REA, augmenté par les événements d'affaires REAAgentIdentification, InstigationEvent et ContractEvent et l'événement informationnel ClaimMaterializationEvent.

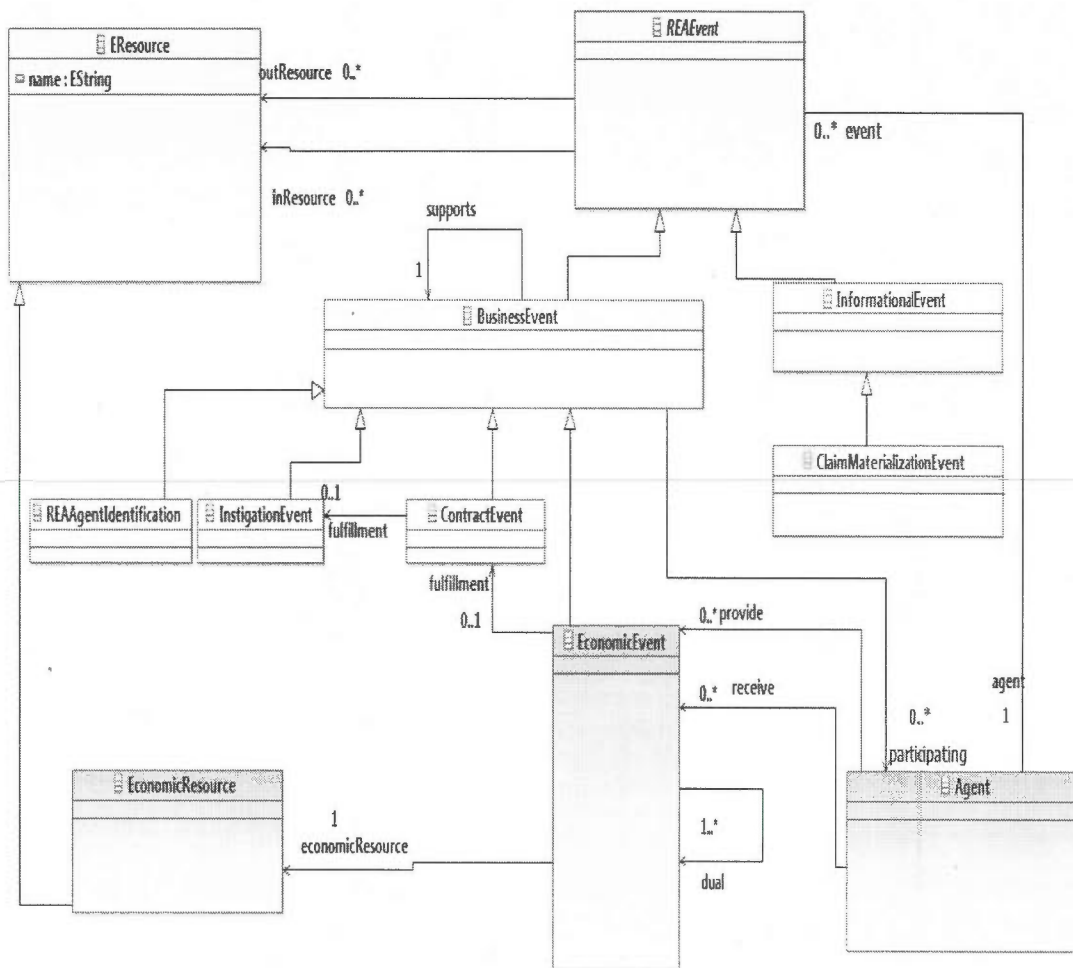


Figure 5.17 Notre méta-modèle REA étendu.

5.2.4 Le méta-modèle comportemental

Le concept d'orchestration est important pour les processus. Il permet de décrire l'ordonnancement, le branchement et la synchronisation des différents événements au sein de la vue dynamique. Nous avons mis en œuvre une partie du modèle comportemental (*Behavior model*) de BPDM. La Figure 5.18 montre l'essentiel de la vue dynamique de notre méta-modèle de processus d'affaires.

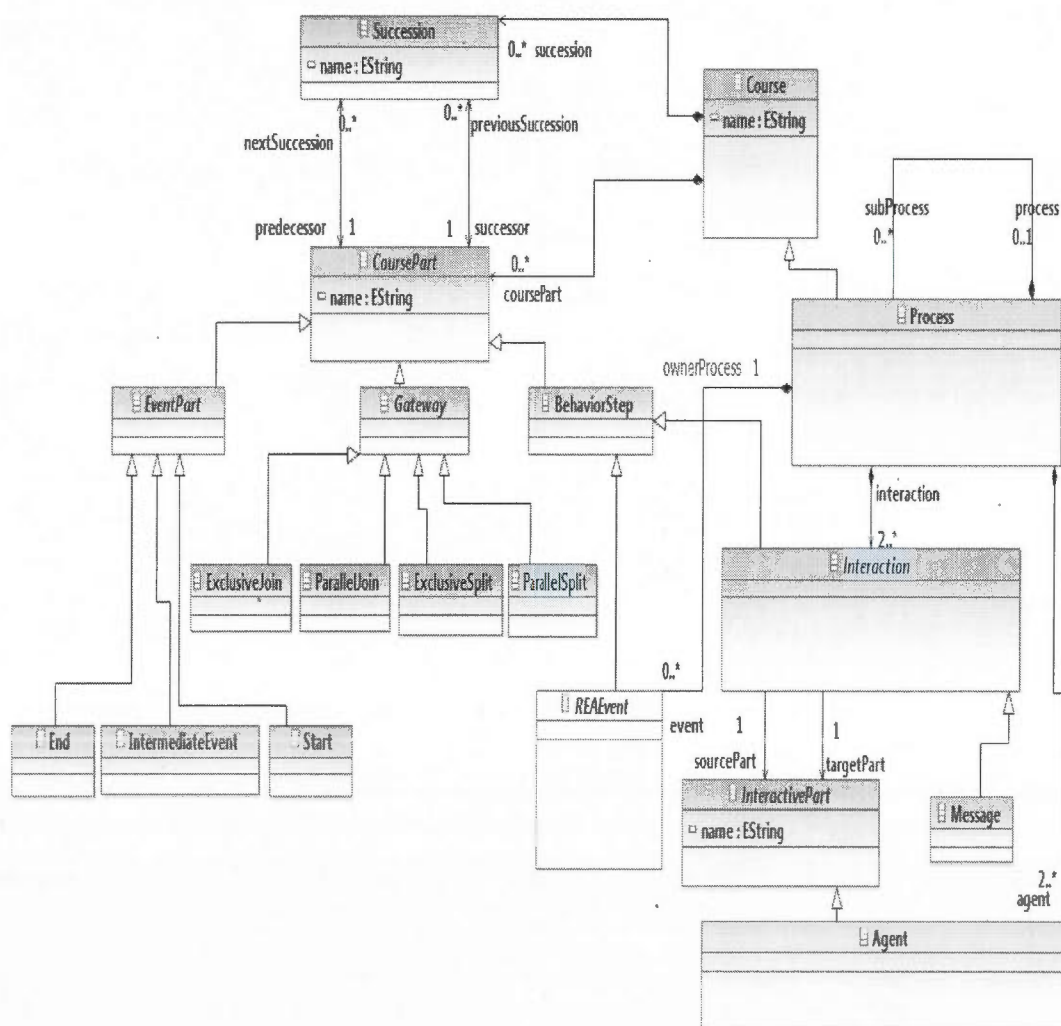


Figure 5.18 Notre méta-modèle de la vue dynamique.

Un processus d'affaires (Process) est un « déroulement » (Course) au sens de BPDM. Il contient un ensemble d'étapes (CoursePart) ordonnées par des successions (Succession). Une étape (CoursePart) du processus d'affaires peut être un événement REA, une interaction (BehaviorStep), un élément de contrôle (Gateway) ou un événement de contrôle (EventPart). Un événement (REAEvent) consomme et produit des ressources (EResource). Un événement REA est soit économique, d'affaires ou informationnel. Les événements sont effectués par des agents REA (Agent) qui jouent certains rôles spécifiques. Notre méta-modèle est présenté à l'appendice C sous forme de diagramme *ecore*.

5.2.5 Implémentation des vues de processus d'affaires

Les vues de processus d'affaires sont des instances du méta-modèle. Chaque vue a ses propres méta-éléments. La vue REA montre les trois types d'événements REA (i.e. économiques, d'affaires et informationnels), les ressources échangées, produites, utilisées et consommées par ces événements ainsi que les agents impliqués dans le processus. La vue dynamique met l'accent sur le concept d'orchestration de processus. La vue informationnelle décrit la structure des entités manipulées par le processus. La vue organisationnelle représente où et par qui les événements sont réalisés. La Figure 5.19 illustre le modèle des vues d'un processus d'affaires.

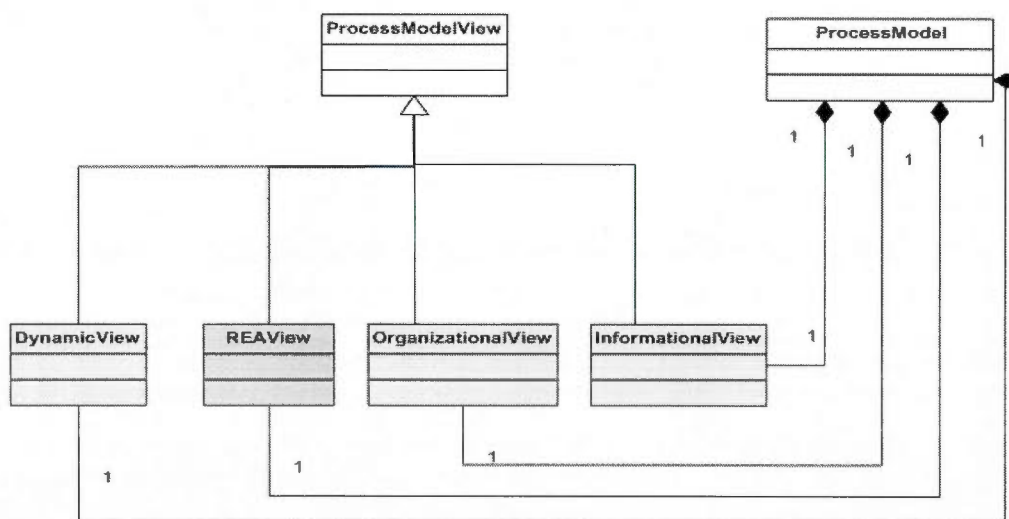


Figure 5.19 Le modèle des vues d'un processus d'affaires.

5.2.6 Exemple : la vue REA du processus de vente et distribution

La Figure 5.20 montre la vue REA du processus de vente et distribution sous forme d'un composant JFace sous Eclipse.

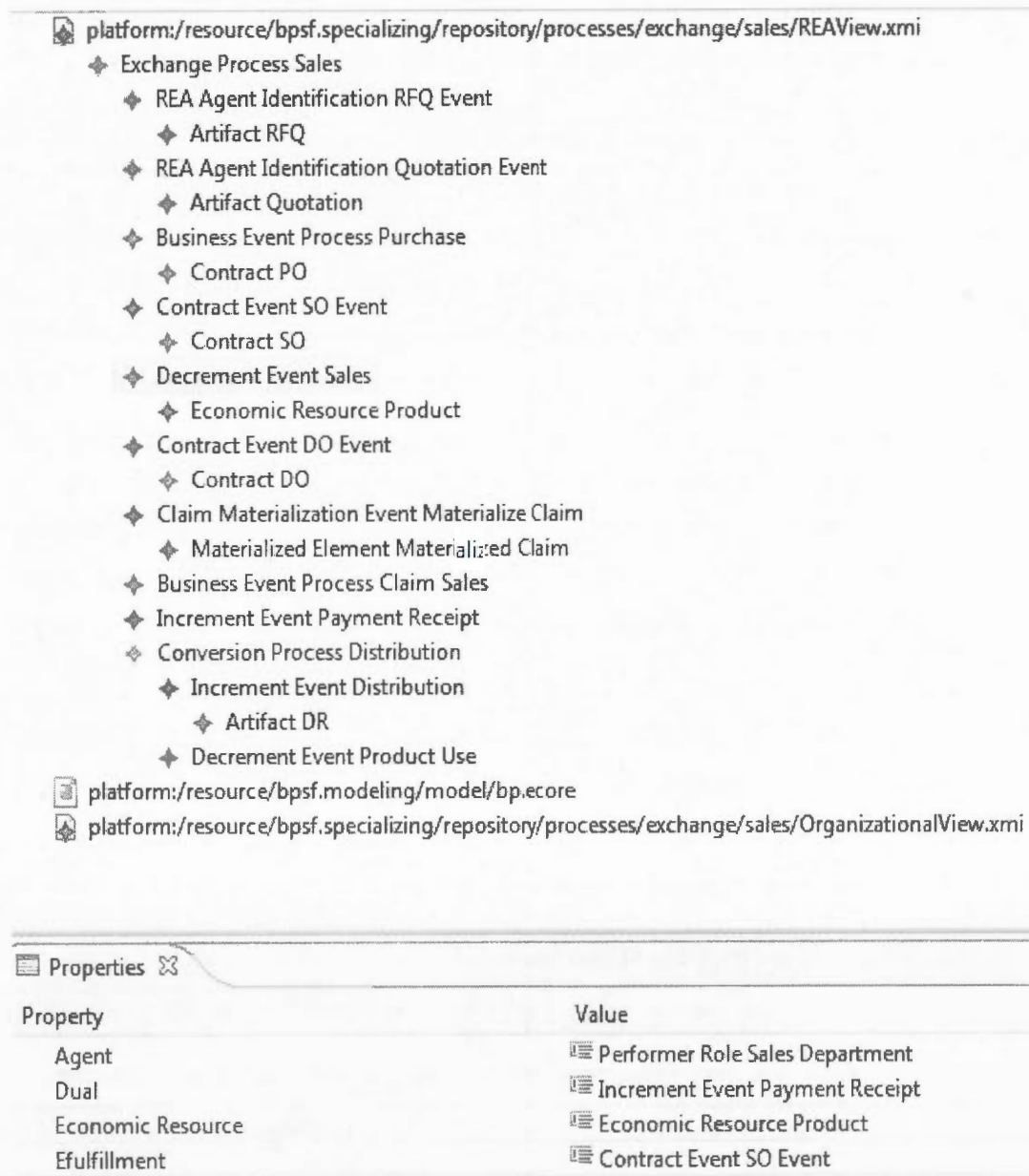


Figure 5.20 La vue REA du processus de vente et distribution.

Cette vue graphique (fig. 5.20) ne montre pas toutes les propriétés des différents éléments de la vue. Par exemple, la relation de dualité n'est pas visible. La vue REA complète est présentée dans l'appendice D sous format XML.

5.3 Vers un catalogue de processus d'affaires génériques

5.3.1 Notre approche de classification de processus génériques

Notre objectif est de fournir un catalogue qui offre un support simple et structuré permettant de se placer sur le modèle de processus le plus proche de ce que nous voulons réutiliser. Pour cela, nous proposons la construction d'un catalogue de processus d'affaires organisé selon une hiérarchie de spécialisation. Dans une hiérarchie de spécialisation, chaque élément hérite des caractéristiques de son parent et les modifie progressivement. Cette structure d'organisation favorise la compréhensibilité, la maintenabilité et la réutilisabilité (Wyner et Jintae, 2003). Selon Malone *et al.* (1999), une hiérarchie de spécialisation offre les avantages suivants : (i) une meilleure compréhensibilité des processus d'affaires, (ii) la génération de modèles de conception alternatifs (i.e. générer d'autres variantes de processus), et (iii) un cadre organisationnel qui permet d'indexer et de localiser facilement les processus. Ce support d'indexation est particulièrement pertinent dans le cas d'un nombre important de processus d'affaires.

Pour faciliter la navigation au sein du catalogue de processus d'affaires (fig. 5.21), nous proposons une classification à deux niveaux. Le premier niveau est descriptif, le deuxième est structurel. La classification descriptive distingue les processus d'affaires horizontaux. Elle est basée sur la fonctionnalité d'affaires. Cette facette décrit la fonctionnalité du processus à un niveau d'abstraction élevé. Elle présente l'objectif du processus. Les initiatives de classification descriptive les plus proches à la nôtre sont celles de Porter (1985), de UN/CEFACT (2005), et des systèmes PGI (Magal et Word, 2010). Les fonctionnalités identifiées sont réparties selon les deux patrons d'échange et de conversion REA. Nous distinguons, d'un côté, la production, l'inspection, la distribution, l'assemblage et la maintenance pour les processus de conversion et, de l'autre, la vente, l'achat, la gestion de ressources humaines, la finance et les services, pour les processus d'échange.

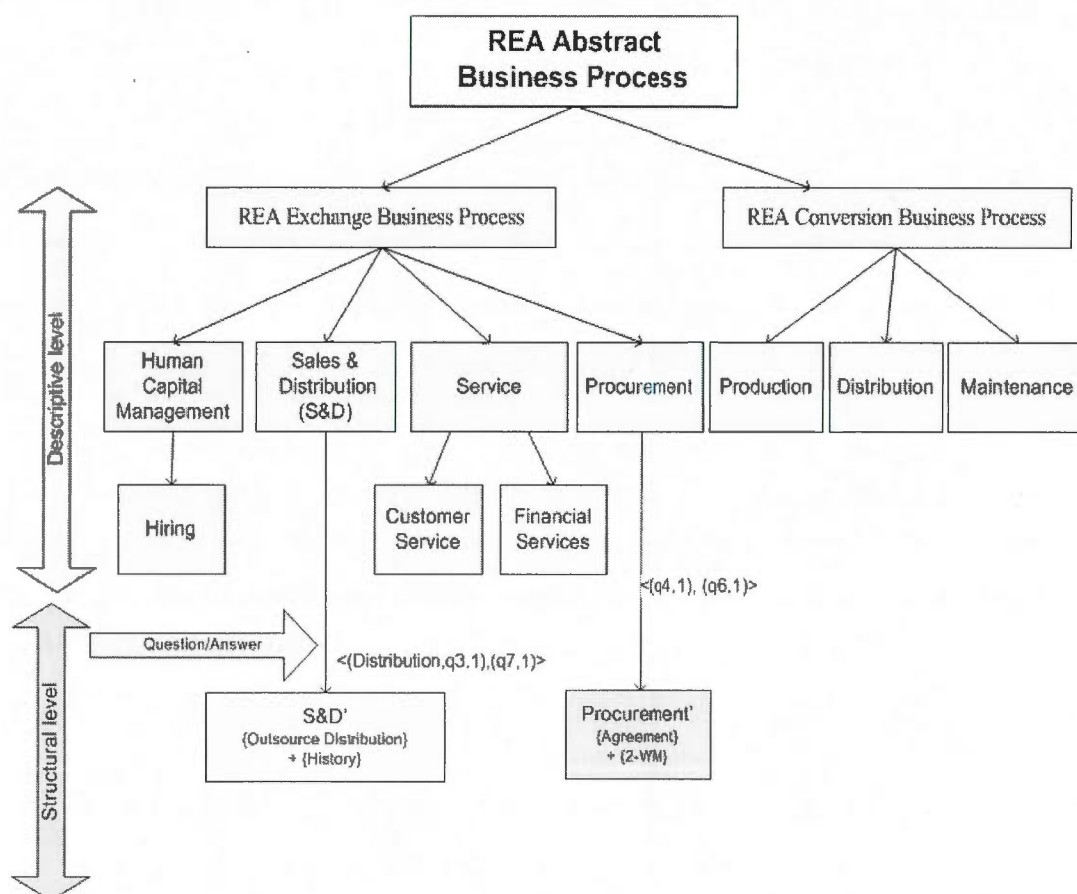


Figure 5.21 La structure de notre référentiel.

Ces catégories de processus d'affaires du niveau descriptif ne sont pas exhaustives. Le catalogue peut être étendu en ajoutant de nouvelles catégories et sous-catégories.

Notre approche de classification structurelle distingue les processus d'affaires verticaux. Elle est basée sur le concept de < question, réponse >. Cette classification est structurelle car elle utilise la structure de notre méta-modèle de processus d'affaires. Cette technique permet de définir le chemin de spécialisation. En effet, le chemin de spécialisation entre deux processus d'affaires P1 et P2 du catalogue est la séquence des couples (question, réponse) spécifiques qui permettent de passer de P1 vers P2. Nous appelons ce couple, l'élément structurel. Lorsque la question s'applique à un processus de la chaîne de valeur, l'élément structurel est un triplet (sous-processus, question, réponse). Par exemple, le chemin

de spécialisation entre le processus *S&D* et le processus *S&D'* est $\langle (\text{Distribution}, q3, 1), (q7, 1) \rangle$. Le premier élément structurel (*Distribution*, *q3*, *1*) indique que l'organisation veut sous-traiter le processus de distribution à un tiers. L'élément (*q7*, *1*) signifie que l'organisation veut garder l'historique des transactions d'échange pour tous les processus de la chaîne de valeur.

5.3.2 Phase de découverte du processus le plus proche

Durant cette phase, l'analyste d'affaires localise un processus d'affaires à spécialiser parmi l'ensemble des processus du catalogue. Le catalogue peut contenir un nombre important de processus d'affaires. Pour cela, l'utilisateur doit clairement identifier les critères de recherche pour pouvoir sélectionner le processus approprié. Pour ce faire, il doit d'abord identifier la classe descriptive. Cette étape est simple puisque la classification descriptive est basée sur la fonctionnalité. Elle peut être réalisée par une simple recherche. Par la suite, il faut faire une recherche basée sur la classification structurelle. Cette recherche utilise le chemin de spécialisation qui est basé sur le concept de $\langle \text{question}, \text{réponse} \rangle$.

La recherche doit permettre de trouver le processus qui répond aux besoins d'affaires ou le processus le plus proche contenant la séquence optimale du chemin de spécialisation.

5.3.3 Mise en œuvre du catalogue

La Figure 5.22 montre la mise en œuvre de notre catalogue de processus d'affaires. Chaque répertoire contient un processus d'affaires spécialisé à partir de son processus parent. Plus précisément, il contient les vues REA, dynamique, organisationnelle et informationnelle du processus. Par exemple, le répertoire (*Assembly*, *q3*, *1*) au dessous du répertoire *production*, contient le processus obtenu après la spécialisation du processus de base de la famille des processus de production suite à une réponse positive à la question de sous-traitance du processus d'assemblage.

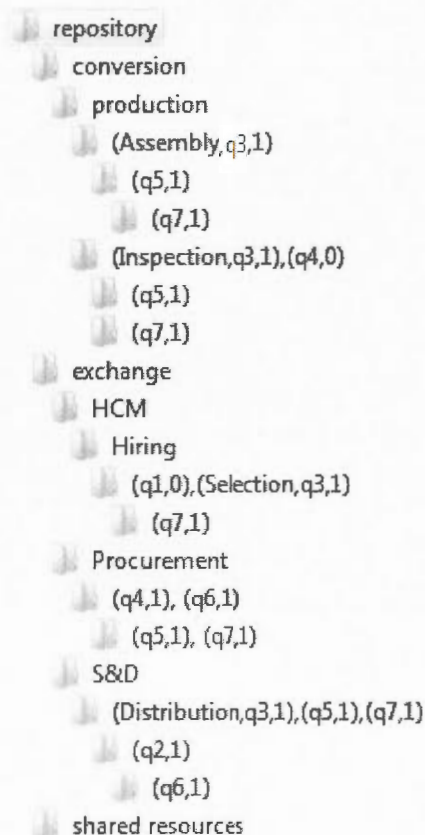


Figure 5.22 Mise en œuvre de notre catalogue de processus d'affaires.

5.4 Conclusion

Nous avons présenté, dans ce chapitre, notre approche pour la représentation de processus d'affaires. Nous avons montré que notre modélisation n'impose pas de stratégie particulière pour la conception du processus d'affaires générique car notre spécialisation supporte les opérations d'ajout, de suppression et de déplacement.

Nous avons décrit, en détail, notre approche de modélisation de processus d'affaires. Cette dernière consiste en deux étapes : une étape d'analyse et une étape de conception. Toutes deux sont basées sur des concepts (e.g. événements, ressources, etc.) et des patrons (e.g. échange, conversion, chaîne de valeur, etc.) de l'ontologie d'affaires REA.

Pour répondre aux besoins des processus d'affaires de nos jours, nous avons augmenté la vue REA avec les événements d'affaires et informationnels. Nous avons aussi augmenté le méta-modèle de l'ontologie REA pour y ajouter le concept d'orchestration en se basant sur le modèle comportemental de la norme BPDm. Finalement, nous avons présenté notre implémentation du méta-modèle de processus d'affaires avec le cadre de modélisation EMF.

Nous avons aussi introduit et décrit, dans ce chapitre, notre approche pour la classification de processus d'affaires. Nous avons proposé la construction d'un catalogue de processus d'affaires à deux niveaux : une classification descriptive basée sur la fonctionnalité d'affaires et une classification structurelle basée sur le concept de < question, réponse >.

Notre catalogue est organisé selon une hiérarchie de spécialisation qui favorise la maintenabilité et la réutilisabilité. Ce type de hiérarchie offre une meilleure compréhension des processus d'affaires, permet la génération de modèles de conception alternatifs et offre un cadre organisationnel qui permet d'indexer et de localiser facilement les processus.

CHAPITRE VI

TRANSFORMATION DE PROCESSUS D’AFFAIRES

L’objectif principal de notre travail est de développer une nouvelle méthodologie et des outils pour la réutilisation et l’adaptation de processus génériques selon les besoins spécifiques des organisations. Nous avons proposé une approche mixte qui combine le catalogage et la génération de processus d’affaires. Dans le chapitre 5, nous avons présenté notre approche d’organisation et de représentation de processus d’affaires. Notre représentation comprend quatre vues : la vue REA étendue, la vue dynamique, la vue informationnelle et la vue organisationnelle. Dans ce chapitre, nous proposons un ensemble de techniques permettant de spécialiser un processus générique pour l’adapter à des besoins spécifiques.

La spécialisation de processus d’affaires que nous préconisons utilise une approche transformationnelle. Nos transformations sont génériques. Elles sont basées sur des patrons d’affaires et des questions. La section 6.1 illustre notre approche d’identification de questions génériques. Nous montrons différents patrons d’affaires ainsi que les questions génériques qui leurs sont reliées. Dans la section 6.2, nous présentons notre approche transformationnelle de processus d’affaires que nous illustrons par un exemple. La section 6.3 décrit notre mise en œuvre des transformations élucidées. La section 6.4 discute quelques problèmes d’ordre transformationnel auxquels nous avons été confrontés.

6.1 Une nouvelle approche de spécialisation par les questions

Nous visons une approche transformationnelle basée sur les questions. Cette approche n’est pas nouvelle. Nous avons présenté quelques initiatives de spécialisation de processus d’affaires

par les questions dans la section 2.4 (voir chap. 2). Les questions utilisées par ces approches posent problème. En effet, elles sont soit liées aux domaines d'affaires spécifiques (e.g. Fowler, 1997; Wohed, 2000; Eriksson et Penker, 2000), soit trop génériques (e.g. Coad, North et Mayfield, 1997) auquel cas, elles ne sont plus réutilisables, ou perdent leur sémantique après la généralisation (e.g. Wohed, 2001). La clé de notre approche est : (i) de trouver une façon de généraliser les questions qui préserve la sémantique métier, et (ii) de généraliser les transformations pour qu'elles puissent s'appliquer à différents domaines d'affaires. Dans cette section, nous montrons notre stratégie d'identification des questions génériques avec le cadre ontologique REA. Les transformations seront présentées dans la section 6.2.

6.1.1 Identification des questions génériques

Pour construire l'ensemble de nos questions génériques, nous avons utilisé principalement la sémantique de l'ontologie d'affaires et de patrons d'affaires REA. Pour les patrons d'affaires, nous avons choisi ceux proposés par Pavel Hruby (2006). Souvent, les questions peuvent aussi être identifiées à partir de la structure du patron.

En parcourant le catalogue de patrons d'affaires REA, nous avons identifié plusieurs questions. Chaque question peut être instanciée pour différents processus d'affaires afin d'obtenir des questions spécifiques au processus. Pour chaque couple (question, réponse) (i.e. l'élément structurel), nous avons identifié des transformations génériques. Dans la suite de cette section, nous énumérons différents patrons d'affaires ainsi que des questions génériques qui leurs sont reliées.

6.1.2 Les patrons REA et les questions génériques

L'ontologie REA distingue deux patrons d'affaires de base, soit le patron d'échange et le patron de conversion. D'autres travaux ont identifié divers patrons d'affaires à partir de l'ontologie REA. Notamment, Pavel Hruby (2006) qui propose une vingtaine de patrons qu'il classifie en patrons structuraux et comportementaux.

Nous avons expliqué, dans le chapitre 5, que notre approche considère les événements économiques, d'affaires et informationnels. Parmi les événements informationnels, nous ne considérons que ceux qui ont une incidence sur le processus d'affaires. Plus précisément, nous

conservons ceux ayant une incidence sur le déroulement d'un ou plusieurs événements d'affaires ou économiques. Dans la même perspective, nous ne considérons pas les questions qui ne spécialisent que la vue informationnelle. Ce type de transformations (spécialisation/extension de données) est relativement bien connu et traité dans les approches orientées objet. Nous faisons ainsi le choix de nous concentrer sur des transformations plus complexes ou celles affectant plusieurs vues du processus d'affaires. Parmi les patrons comportementaux (selon la classification de Hruby) qui sont exclus, nous citons le patron de réconciliation des créances (*Reconciliation pattern*), le patron de classification des entités REA (*Classification pattern*) et le patron d'informations agrégées (*Account pattern*).

Dans la section suivante, nous présentons quelques exemples de patrons d'affaires et les questions génériques correspondantes.

6.1.2.1 Le patron d'engagement

Un engagement économique (*Commitment*) est une promesse d'échange ou de conversion. Un engagement représente une obligation future d'un événement économique. Nous avons discuté du concept d'engagement et de sa relation avec l'événement économique dans la section 4.3 (voir chap. 4). Dans cette section, nous présentons le patron d'affaires de l'engagement économique au complet (fig. 6.1).

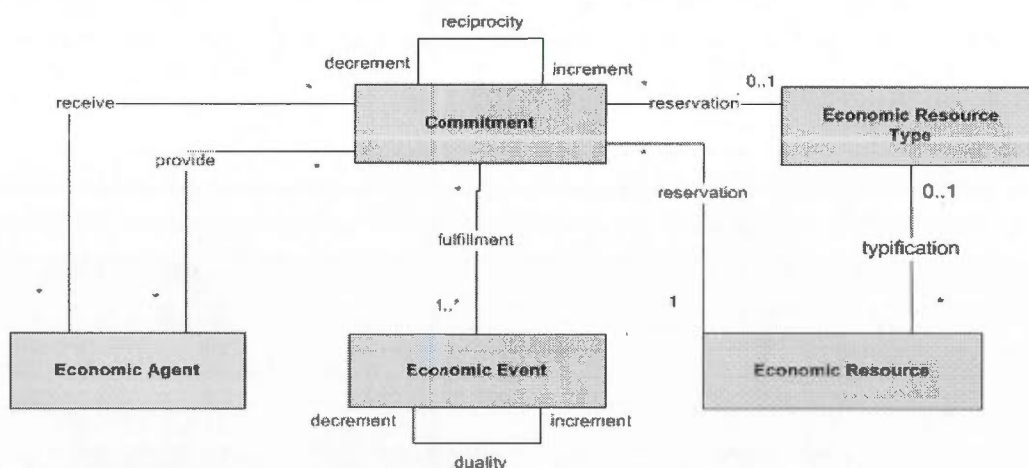


Figure 6.1 Le patron d'affaires « Engagement ». (Source : Hruby, 2006)

Chaque processus REA distingue deux types d'engagements : un engagement d'incrément (*Increment Commitment*) et un engagement de décrétement (*Decrement Commitment*). L'engagement d'incrément doit être accompli par l'événement économique d'incrément, tandis que l'engagement de décrétement doit être accompli par l'événement économique de décrétement. Ces deux types d'engagements sont reliés par la relation *reciprocity*. La relation *fulfillment* permet de valider si les engagements ont été exécutés par les événements économiques associés.

L'engagement peut être lié à la ressource économique concrète ou à son type par la relation *reservation*. En effet, dans la plupart des processus d'affaires, les engagements réservent des types de produits. Finalement, les relations *provide* et *receive* entre l'engagement et l'agent économique ont la même sémantique que celles liant l'événement économique et l'agent économique.

En utilisant notre stratégie d'identification des questions génériques, nous pouvons utiliser la sémantique des engagements REA pour déterminer si l'échange est régi par des obligations futures. S'il n'y a pas d'obligations futures, les événements duaux sont instantanés. Dans ce cas, le contrat REA n'est pas nécessaire. Dès lors, nous envisageons la question suivante : *Est-ce que le déroulement du processus d'affaires {0} est régi par des engagements ?* Où {0} est le paramètre positionnel qui fait référence au processus REA. Aussi, en se basant sur ce même patron ou le patron d'affaires *Type (Typification)* entre la ressource économique et son type, l'engagement économique peut être basé sur le type d'un produit et non le produit lui-même. Nous pouvons alors envisager la question : *Est-ce que les engagements économiques du processus {0} sont basés sur des ressources économiques concrètes ?*

6.1.2.2 Le patron de contrat

Un contrat est un accord entre deux ou plusieurs parties. Un contrat commercial a pour objectif l'établissement d'engagements entre des parties pour l'échange de ressources (e.g. de l'argent, des biens, des services, etc.) (Andersson *et al.*, 2006). Nous avons présenté le concept de contrat dans la section 4.3 (voir chap. 4). Ce dernier a été introduit comme une extension ontologique de REA par Geerts et McCarthy (2000). Nous avons montré la relation entre le

contrat, les engagements (*Commitments*), et les termes (*Terms*). En effet, le contrat permet de régler le comportement entre les organisations en définissant :

- (i) les promesses d'échange ou de conversion, dites clauses du contrat, et
- (ii) les termes du contrat qui représentent des obligations spéciales à exécuter sous certaines conditions (e.g. la violation de certaines promesses).

Le bon de commande, dans un processus d'approvisionnement, est un exemple de contrat entre l'organisation et le fournisseur. Les lignes du bon de commande sont des engagements. On distingue deux types d'engagements : les lignes d'achat et les lignes de paiement. Les lignes d'achat représentent les engagements pour recevoir les ressources économiques. Les lignes de paiement représentent les engagements de payer ces ressources économiques. La Figure 6.2 illustre le patron contrat tel que proposé par Hruby (2006).

Le concept de contrat est important à modéliser car les agents économiques qui se sont entendus pour l'échange ou la conversion ne sont pas nécessairement les agents fournisseurs et receveurs des événements économiques au sens REA. Un contrat peut lier de deux à plusieurs agents via la relation *party*. Les agents reliés au contrat ne sont pas nécessairement les mêmes qui sont reliés aux engagements du même contrat, qui à leur tour, ne sont pas nécessairement les agents fournisseurs et receveurs qui exécutent les obligations.

L'accord (*Agreement*) économique représente un niveau plus élevé de contrat entre les agents économiques. L'accord régleme le comportement des contrats individuels. Par exemple, nous pouvons citer l'accord de service pour la maintenance d'équipements qui consiste à offrir des contrats de maintenance sous certaines conditions. Notons qu'un accord peut être réglemé par un autre accord plus haut dans la hiérarchie.

En utilisant la sémantique de ce patron d'affaires, nous pouvons examiner si l'échange (ou la transformation) de ressources implique un accord. Dès lors, nous envisageons la question suivante : *Est-ce que le déroulement du processus {0} est régi par un accord ?*

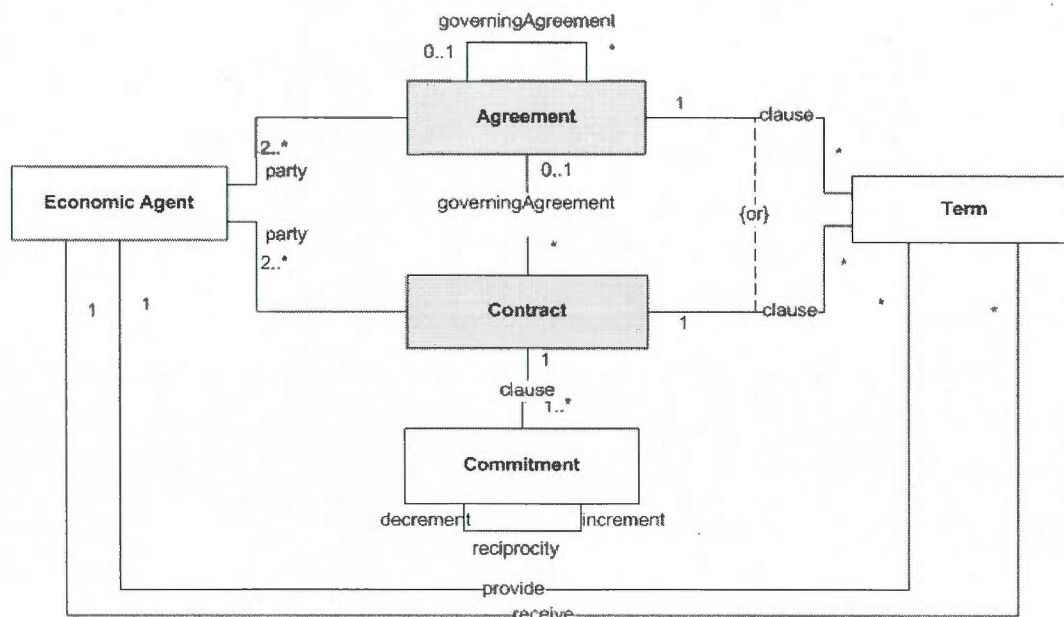


Figure 6.2 Le patron d'affaires « Contrat ». (Source : Hruby, 2006)

6.1.2.3 Le patron de sous-traitance

La sous-traitance d'un processus d'affaires consiste à confier à une organisation tierce (le sous-traitant), la réalisation de ce processus d'affaires. En général, les organisations ont recours à la sous-traitance des processus internes non stratégiques pour se concentrer sur les processus essentiels. En REA, on peut seulement échanger des ressources économiques ; on ne peut pas échanger des processus d'affaires (Hruby, 2006). Lorsqu'un agent économique réalise un processus d'affaires pour un autre agent, les deux agents économiques échangent un service via un nouveau processus d'achat. Le service acheté est une ressource transitoire qui est consommée aussitôt créée. La sous-traitance de processus d'affaires est valable pour les processus de conversion (e.g. transport, production, etc.) et les processus d'échange (e.g. achat, ressources humaines, etc.).

Nous présentons en détail le patron de sous-traitance dans le cas d'un processus de conversion. Le même patron s'applique de façon analogue aux échanges à une différence près. En effet, dans le processus de conversion, la nouvelle ressource économique (i.e. le service

acheté) sera consommée et jumelée à l'autre événement de décrémentation via la relation de dualité de conversion. Cependant, dans le cas d'un processus d'échange, le service acheté sera acquis et jumelé à l'autre événement de décrémentation via la relation de dualité d'échange. Cette analyse se compare aisément à la conception REA de la ressource économique « main-d'œuvre » dans le cas de processus d'échange et de conversion. Effectivement, la « main-d'œuvre » est utilisée via la relation *use* dans le cas d'un processus de conversion, alors qu'elle est acquise via la relation *inflow* dans le cas d'un processus d'échange. Notons que cette analyse reste la même pour l'internalisation, où le service est acheté mais exécuté à l'interne par des agents externes.

La Figure 6.3 montre la chaîne de valeur du processus de conversion qu'on veut sous-traiter. La ressource économique R2 peut désigner la même ressource en entrée (i.e. R1) dont certains attributs ont été altérés ou, désigner une nouvelle ressource créée par la conversion.

Après la sous-traitance du processus, la nouvelle ressource économique (i.e. le service acheté) sera consommée dans le processus de conversion. L'événement de consommation du service sera jumelé à l'événement de décrémentation initial pour réaliser la conversion via la relation de dualité avec l'événement d'incrément. Cette analyse est valide dans les deux cas de conversion, soit la création de nouvelles ressources économiques ou le changement d'attributs des ressources économiques existantes par l'utilisation ou la consommation d'autres ressources. La Figure 6.4 montre la nouvelle chaîne de valeur après la sous-traitance du processus initial (fig. 6.3).

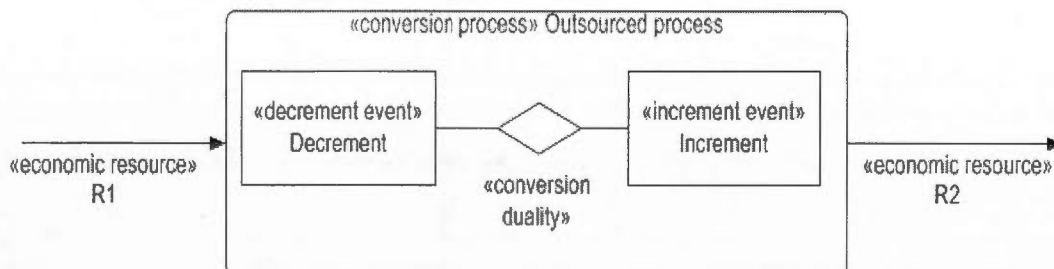


Figure 6.3 La chaîne de valeur du processus de conversion initial.

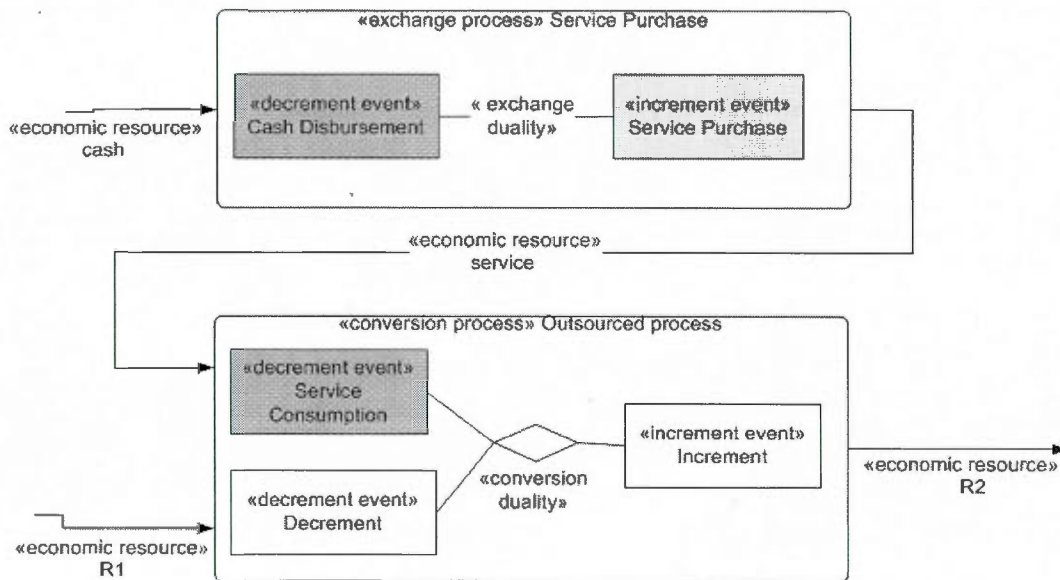


Figure 6.4 La nouvelle chaîne de valeur après la sous-traitance.

La Figure 6.5 montre la vue REA du processus initial (fig. 6.3). Après la sous-traitance, un nouveau processus d'échange est créé. La Figure 6.6 montre la vue REA du processus final après la sous-traitance. Nous allons illustrer tous ces concepts de la sous-traitance en REA par un exemple détaillé dans la section 6.2.

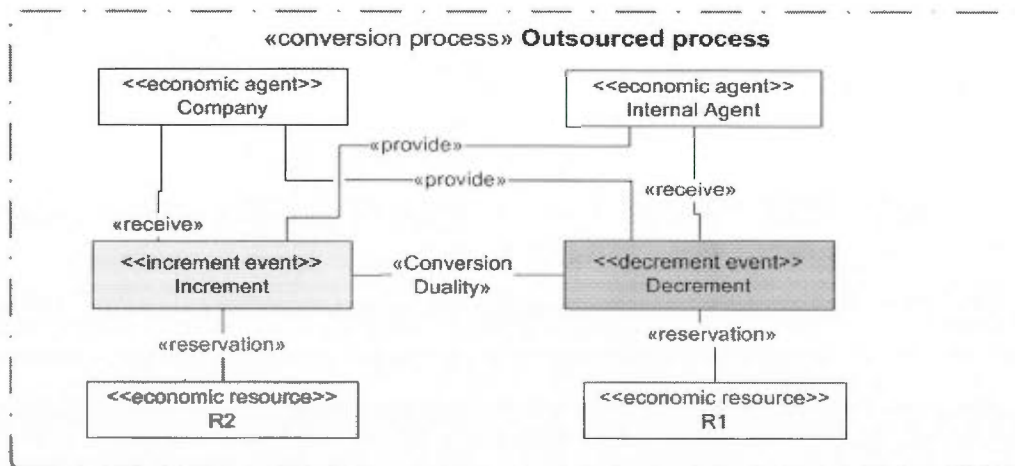


Figure 6.5 Le modèle REA de base du processus de conversion initial.

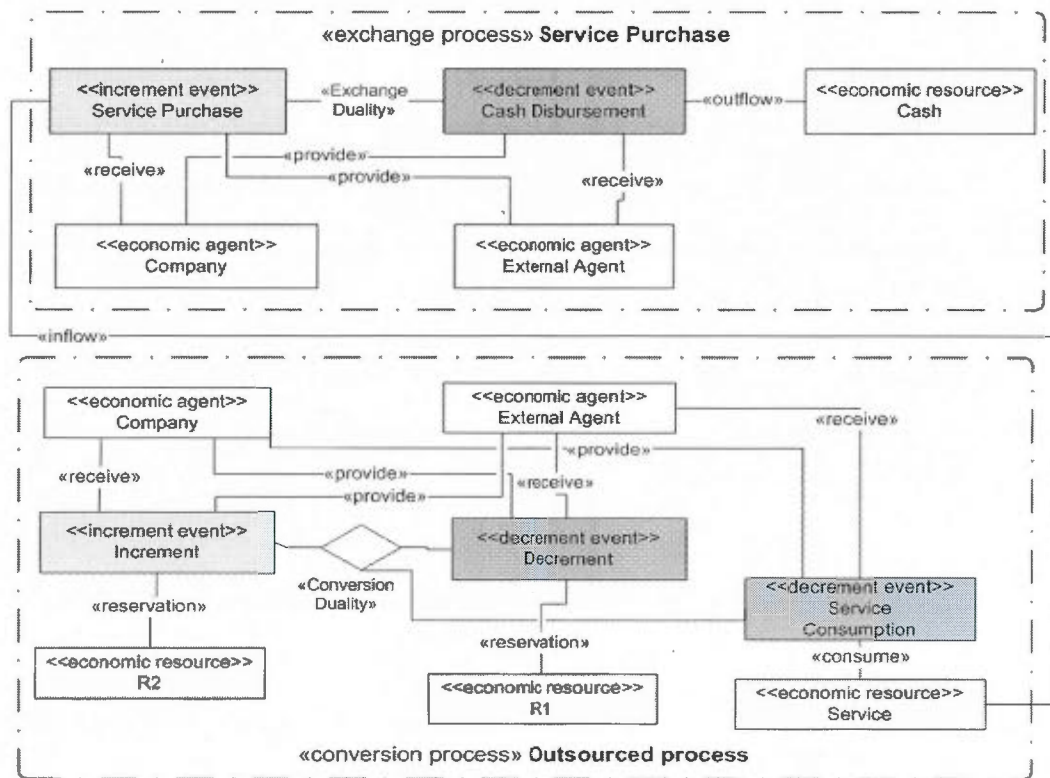


Figure 6.6 Le modèle REA final après la sous-traitance.

6.1.2.4 Le patron de matérialisation de la créance

Il arrive souvent que les événements économiques duaux intervenants dans un processus d'échange ne se produisent pas simultanément. Dans ce cas, un déséquilibre au niveau des ressources économiques échangées est créé durant le laps de temps entre les événements économiques duaux. Pour répondre à cette problématique, McCarthy (1982) a introduit le concept de créances (*Claims*) entre les événements économiques. Ce concept est illustré dans la Figure 6.7.

En général, l'agent économique dont les ressources REA ont été décrémentées informe l'agent partenaire dans l'échange de la valeur du déséquilibre par l'envoi d'un rapport qui représente une matérialisation de la créance (e.g. facture). Ce rapport peut être obtenu par le patron d'affaires *Materialized Claim*. La Figure 6.8 montre le patron de matérialisation de la créance REA tel que proposé par Hruby (2006).

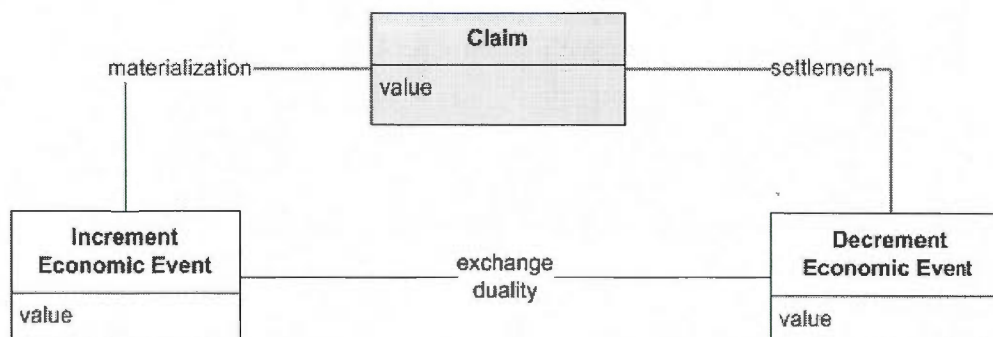


Figure 6.7 Le concept de créance. (Source : Hruby, 2006)

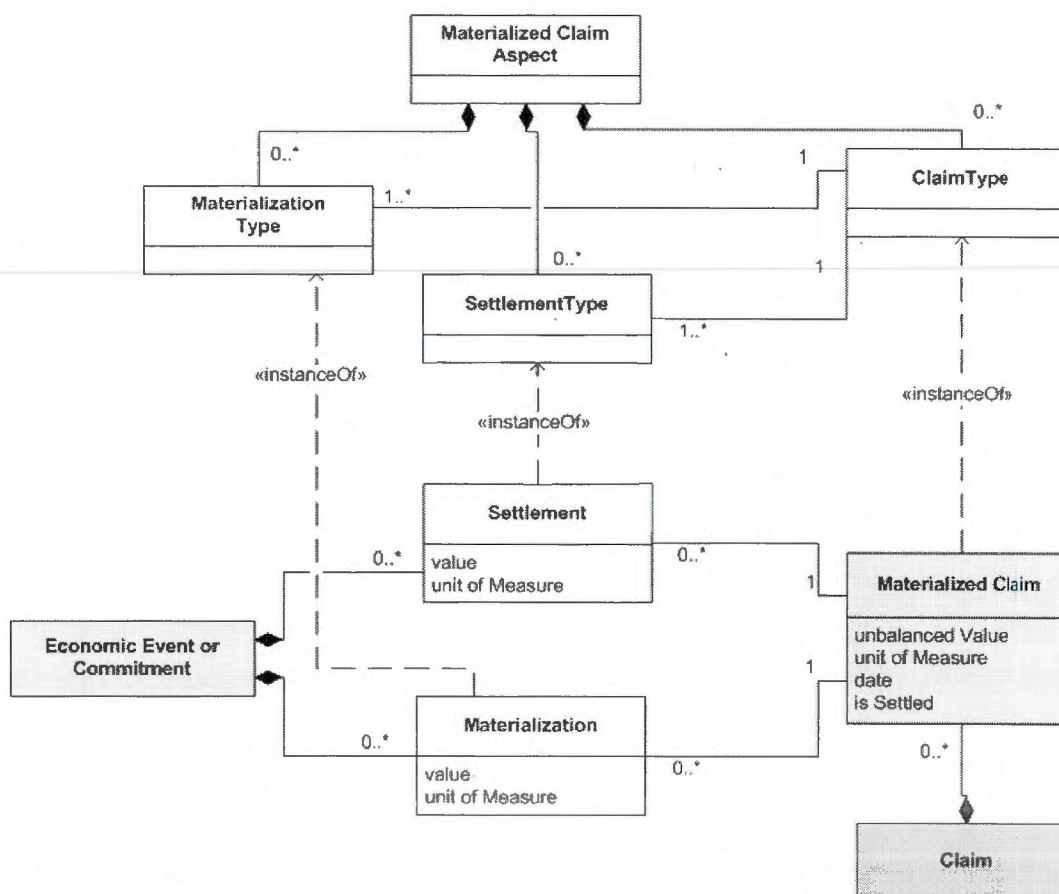


Figure 6.8 Le patron d'affaires « Matérialisation de créance ». (Source : Hruby, 2006)

Lorsque les attributs *value* des engagements réciproques sont comparables, la valeur de la créance peut être calculée automatiquement. Cette valeur peut être obtenue par la différence entre les valeurs des engagements, c'est-à-dire les engagements d'incrément et de décrément. Dans ce cas, l'agent économique dont les ressources REA ont été décrémentées n'est pas tenu d'informer l'agent partenaire par l'envoi de rapport de matérialisation. L'agent partenaire peut alors compléter l'échange en calculant la valeur de la créance automatiquement. Cette méthode se base sur le principe du *Two-way match* qui consiste en un calcul utilisant seulement le contrat REA et le bon de réception des ressources économiques. S'il n'y a pas de variation, l'échange sera complété. De là, nous envisageons la question suivante : *Est-ce que le processus {0} supporte le concept de Two-way match pour la gestion des créances ?*

Dans le cas d'une réponse positive, l'événement de matérialisation de la créance et d'envoi du rapport sera supprimé et le processus qui termine l'échange sera modifié. En effet, plutôt que d'utiliser la validation classique du type *Three-way match* qui se base sur trois composants (i.e. la créance, le contrat, et le bon de réception des ressources économiques), le processus calcule la valeur de la créance automatiquement en se basant seulement sur le principe du *Two-way match* qui se base seulement sur deux composants : le contrat et le bon de réception des ressources économiques. Nous rappelons que cette stratégie de validation a été développée par la compagnie *General Motors* en 1994 (Murray, 2009).

6.1.3 Représentation des questions

Les questions génériques sont applicables à différents modèles de processus d'affaires. Comme pour l'approche métalangage, nous considérons une question générique comme une fonction qui prend des paramètres et retourne une valeur. Les types des paramètres sont des entités du méta-modèle. Par exemple, dans la question : *Est-ce que le processus d'affaire {0} supporte le concept de Two-way match pour la gestion des créances ?*, le paramètre positionnel {0} fait référence à la méta-classe *ExchangeProcess*. La valeur de retour de la fonction est la réponse à la question. Prenons l'exemple de la question générique suivante : *Est-ce que le déroulement du processus {0} est régi par un accord ?* On peut considérer cette question comme étant la fonction suivante :

```
boolean isThereAContractThatGovernsTheProcess(Process p) (1)
```

Notez que cette même question a été formulée comme suit lors de notre première approche :

boolean processExecutionFollowsContractBetween(Actor a, Actor b).(2)

où <Actor> est un élément du méta-modèle représentant un acteur.

La nouvelle question présente un avantage important par rapport à la question de l'approche métalangage quant à la sémantique d'affaires et au concept de filtrage. En effet, un contrat ou un accord n'implique pas toujours deux acteurs. Il peut être conclu entre plusieurs acteurs. De plus, ces acteurs ne sont pas nécessairement ceux impliqués dans le processus d'échange. Quant au filtrage, l'instanciation de la question (2) peut produire un nombre important de questions spécifiques. Par exemple, dans notre exemple de processus d'approvisionnement du chapitre 3 (fig. 3.16), nous retrouvons trois acteurs : « Requirer », « Supplier » et « Accounting ». L'ensemble des questions spécifiques contient toutes les combinaisons possibles d'association de ces trois paramètres. Nous obtenons alors neuf questions spécifiques dont plusieurs sont clairement non pertinentes. Avec la question (1), nous obtenons une seule question.

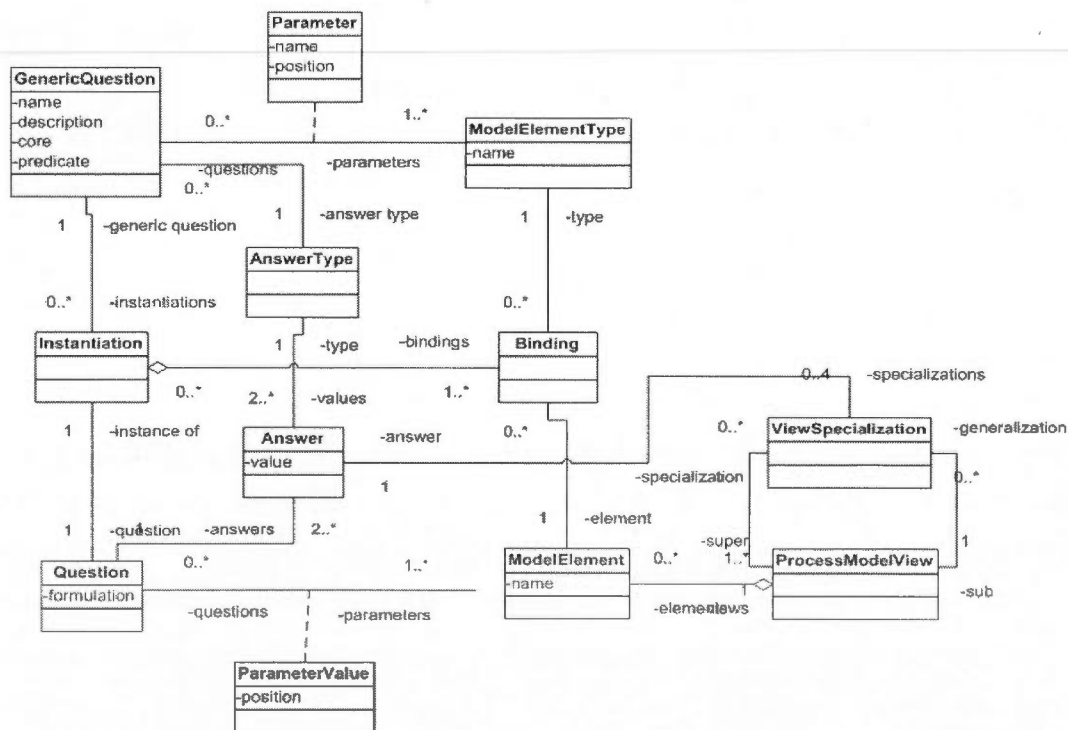


Figure 6.9 Notre modèle de représentation des questions.

Notre représentation des questions génériques comprend un attribut *predicate* qui permet de définir les conditions d'applicabilité de la question. La structure du modèle est similaire à celle de l'approche métalangage. La Figure 6.9 montre notre modèle de représentation des questions génériques et spécifiques.

6.1.4 Implémentation des questions

Nous avons implémenté notre méta-modèle de question avec un schéma XML. La Figure 6.10 montre la représentation de la question : *Est-ce que le déroulement du processus {0} est régi par un accord ?*

L'attribut *Id* est l'identificateur de la question générique. La balise `<core>` définit le texte en langage naturel de la question. `{0}` désigne le paramètre positionnel de la question. L'attribut `answerType` de la balise `<core>` spécifie le type de la réponse spécifique. Les paramètres de la question se trouvent entre les balises `<parameters>`.

```
<!-- ***** -->
<!-- Contract pattern -->
<!-- ***** -->
<question id="q4">
  <name>AGREEMENT_IN_THE_PROCESS</name>
  <description>
    Contract regulates the behavior among organizations and individuals.
    A contract is an entity containing commitments and terms. Agreement
    is a higher level contract that regulates the behavior of individual
    contracts. SLA is an example of agreement.
  </description>
  <core answerType="AnswerType.BOOLEAN" >
    Is there an agreement that governs the business process {0}?
  </core>
  <parameters >
    <parameter>Process</parameter>
  </parameters>
</question>
```

Figure 6.10 Exemple de représentation d'une question générique.

6.2 Les transformations de spécialisation

6.2.1 Principes

La spécialisation d'un processus d'affaires pour une organisation permet de l'adapter à ses besoins d'affaires spécifiques. Notre approche de spécialisation de processus d'affaires consiste à appliquer une série de transformations à ce dernier. Plus particulièrement, les transformations s'appliquent aux vues des processus d'affaires qui composent la chaîne de valeur.

Dans cette section, nous rappelons d'abord les vues d'un processus d'affaires. Nous décrivons, ensuite, notre approche de spécialisation ainsi que les règles de transformation des vues. Finalement, nous discutons de la dépendance entre les vues et de la problématique reliée à l'ordre d'exécution des transformations. Les problèmes reliés à d'autres aspects transformationnels seront présentés dans la section 6.4.

6.2.1.1 Rappel de la représentation des quatre vues

Nous avons vu, dans le chapitre 5, qu'un processus d'affaires est représenté par quatre vues : (i) la vue REA étendue montre les événements REA (i.e. économiques, d'affaires et quelques événements informationnels), les ressources manipulées par ces événements et les agents impliqués, (ii) la vue dynamique décrit l'orchestration au sein du processus d'affaires, (iii) la vue organisationnelle représente la structure organisationnelle, les rôles et les mécanismes de communication au sein des organisations, et (iv) la vue informationnelle modélise les ressources manipulées par les événements du processus d'affaires. Ainsi, nous proposons de représenter chaque processus d'affaires par un quadruplet $\langle \text{REA}, D, O, I \rangle$ où :

- REA : représente la vue REA étendue.
- D : représente la vue dynamique.
- O : représente la vue organisationnelle.
- I : représente la vue informationnelle.

6.2.1.2 Le processus de spécialisation

REA considère le processus d'affaires comme une séquence de patrons d'échange et de conversion de ressources. De là, nous nous sommes concentrés sur les patrons d'affaires REA comme unité de spécialisation. En effet, ces patrons d'affaires sont l'objet de nos questions génériques (sect. 6.1) et, aussi, de nos opérateurs de spécialisation. Dans ce contexte, on s'est posé la question suivante : *comment allons-nous d'un processus d'approvisionnement générique, par exemple, à des variantes spécialisées en se basant sur les réponses d'un analyste d'affaires à des questions posées par notre système ?* L'approche que nous proposons va comme suit : à partir de

- (i) un catalogue patrons d'affaires,
- (ii) un catalogue de questions génériques relatives à chaque patron d'affaires,
- (iii) un catalogue d'opérateurs de spécialisation correspondant à des réponses différentes pour chaque patron, et
- (iv) un processus d'affaires sélectionné depuis un référentiel et conçu sous forme d'une chaîne de valeur de processus REA;

il faut :

- a) instancier les questions correspondantes à différents patrons applicables au processus en entrée,
- b) fournir des réponses, et
- c) appliquer les opérateurs de spécialisations génériques correspondants à chacune des réponses pour chaque vue du processus d'affaires de la chaîne de valeur.

Le processus BPMN de la Figure 6.11 montre les étapes de spécialisation d'un processus d'affaires générique.

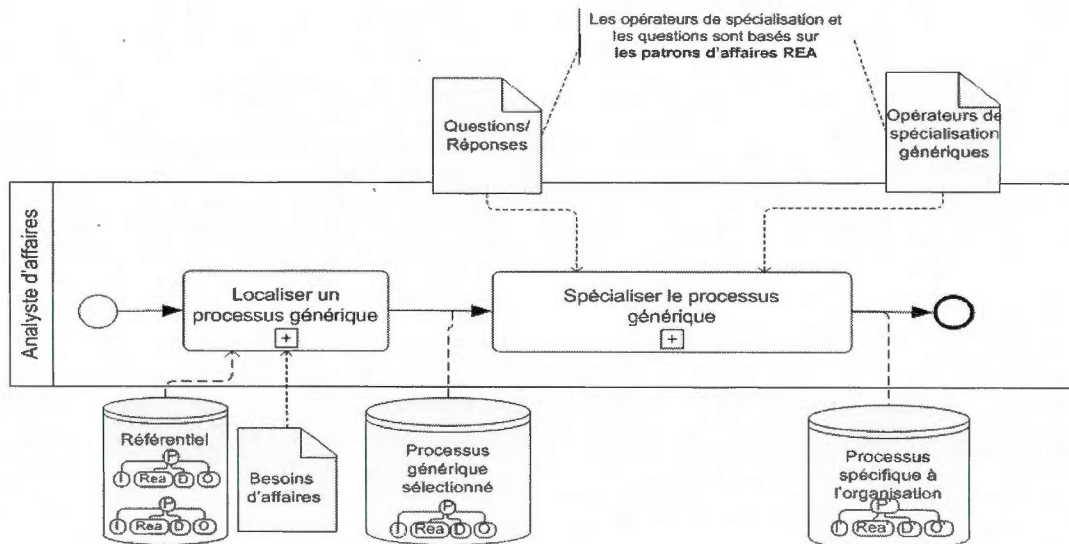


Figure 6.11 Le processus de spécialisation.

6.2.1.3 Les règles de transformation

La spécialisation est l'application d'une série de transformations génériques à un processus d'affaires. À partir de notre modèle de questions (fig. 6.9), chaque réponse à une question spécifique peut transformer zéro, une ou plusieurs vues d'un processus d'affaires. Plus précisément, la transformation dépend du triplet (question, réponse, vue).

Nous avons appelé le couple (question, réponse) l'élément structurel (voir sect. 5.3). Ce dernier devient un triplet (sous-processus, question, réponse) lorsque la question s'applique à un sous-processus de la chaîne de valeur. La transformation associée à un seul élément structurel (i.e. après une réponse à une seule question) est dite *unitaire*. Quand l'élément structurel transforme les quatre vues d'un processus d'affaires, nous disons qu'il est *maximal* et que la transformation unitaire correspondante est *maximale*. Finalement, la transformation associée à plusieurs éléments structurels est dite *globale*. Cette dernière consiste en l'application d'une série de règles de transformation associées à plusieurs questions/réponses. Dans la suite de ce document, nous utiliserons simplement le terme « *transformation* » pour

désigner une transformation unitaire. Nous allons utiliser ces concepts pour expliquer les règles de transformation.

Un processus spécialisé est le résultat de l'application d'une séquence de transformations à un autre processus plus générique. Si le nouveau processus spécialisé $P' \langle \text{REA}', D', O', I' \rangle$ est obtenu après une série de séquences $\langle (\text{processus}_i, \text{question}_i, \text{réponse}_i) \rangle$ pour $i=1..n$ à partir d'un processus d'affaires $P \langle \text{REA}, D, O, I \rangle$, alors, nous appelons la séquence $\langle (\text{processus}_1, \text{question}_1, \text{réponse}_1), (\text{processus}_n, \text{question}_n, \text{réponse}_n) \rangle$, le *chemin de spécialisation* du processus $P' \langle \text{REA}', D', O', I' \rangle$ à partir de $P \langle \text{REA}, D, O, I \rangle$. C'est ce chemin qui est la base de notre hiérarchie de spécialisation dans le catalogue de processus génériques (voir sect. 5.3).

Pendant le processus de spécialisation, nous ne sauvegardons pas les processus intermédiaires (i.e. après chaque transformation). Seulement le nouveau processus sera ajouté au référentiel ainsi que le chemin de spécialisation qui servira de trace. La Figure 6.12 montre notre modèle de spécialisation de processus d'affaires.

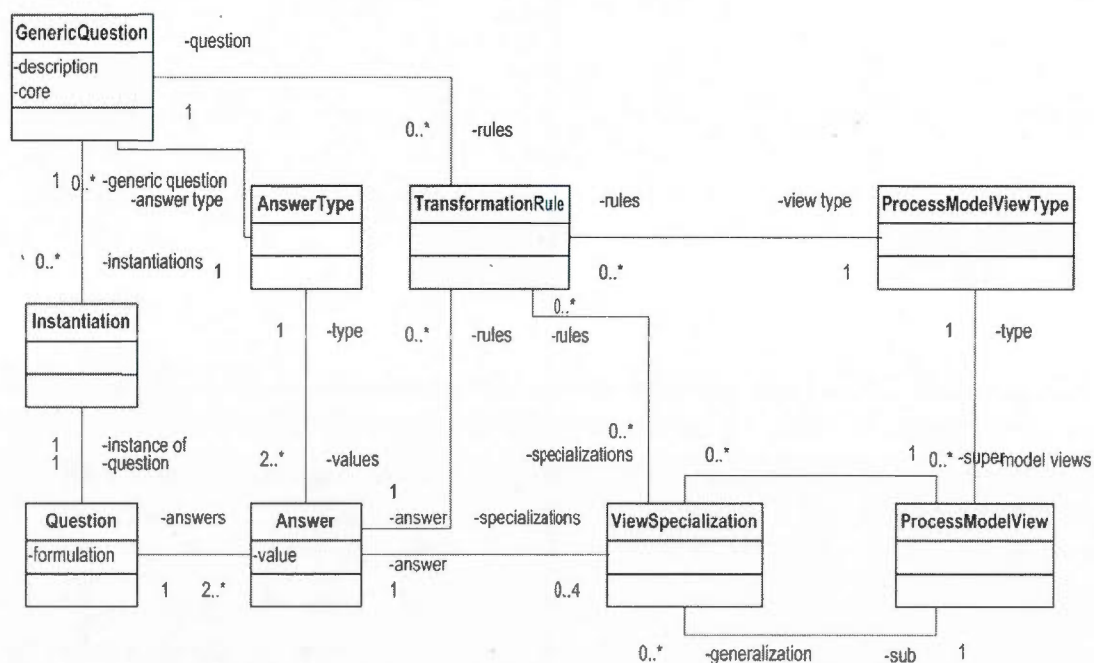


Figure 6.12 Le modèle de spécialisation de processus d'affaires.

Le modèle de spécialisation de la Figure 6.12 montre qu'une réponse à une question spécifique peut donner une variante plus spécifique du processus générique. Cette spécialisation va se refléter de façon différente au niveau de chacune des vues.

À partir des patrons d'affaires REA, nous avons développé plusieurs transformations génériques. Par exemple, pour le patron du contrat, l'existence d'un accord entre les parties du contrat éliminera le besoin de rechercher le partenaire d'affaires (i.e. l'agent économique externe) car ce dernier est déjà connu. Au niveau du modèle, ceci va se manifester, pour chaque vue de manière générique comme suit :

- *Vue REA* : Supprimer les événements REA liés à l'identification de l'agent partenaire. Les ressources utilisées exclusivement pour l'identification du partenaire sont retirées de la vue REA.
- *Vue Organisationnelle* : Supprimer le lien entre les agents et les tâches (i.e. les événements REA) liées à l'identification de l'agent partenaire.
- *Vue Dynamique* : Ajuster les liens comportementaux suite à la suppression des événements REA. Cela se traduit par l'ajout, le déplacement ou la suppression des séquences d'ordonnancement et de messages dépendamment des événements REA supprimés.
- *Vue Informationnelle* :
 - i. Ajouter une nouvelle classe *Agreement* pour représenter l'accord. Ajouter aussi l'entité qui représente le contrat REA si celle-ci n'existe pas¹.
 - ii. Ajouter les associations bidirectionnelles entre cette nouvelle classe et les classes qui représentent les agents économiques selon le patron d'affaires *Contrat*².
 - iii. Ajouter les associations bidirectionnelles entre cette nouvelle classe et l'entité qui représente le contrat REA selon le patron d'affaires *Contrat*.

¹L'accord régleme les contrats individuels. Si on veut ajouter un accord, il faut le lier à l'entité qui modélise contrat REA. Ce dernier peut ne pas exister, dépendamment du processus d'affaires qu'on veut spécialiser.

²Nous rappelons que nous avons décidé de montrer les agents REA impliqués dans le processus d'affaires dans la vue informationnelle afin d'en faciliter la lecture et la compréhension (sect. 3.3).

- iv. Supprimer de la vue tous les éléments informationnels liés exclusivement aux événements d'affaires REA d'identification de l'agent partenaire.

Comme nous pouvons le constater, ces transformations ont peu à voir avec la spécialisation en orienté-objet : dans un cas, on ajoute une classe et des associations, dans l'autre, on enlève des étapes d'un processus. Ceci rejoint, en quelque sorte, les observations de Lubars *et al.* (1992). En effet, Lubars *et al.* (1992) ont observé que, bien que des modifications mineures dans les exigences résultent en général en des modifications mineures au niveau du modèle de classe —l'équivalent de notre vue informationnelle—, ces mêmes modifications pouvaient entraîner des changements importants au niveau des modèles comportementaux tels que la vue dynamique. De là, Mili *et al.* (2009) déduisent (i) que la spécialisation ne correspond pas toujours aux concepts d'extension en orienté objet, et (ii) qu'elle peut être simple pour une vue mais complexe pour une autre vue.

6.2.1.4 L'ordre d'exécution des transformations des vues

Nous avons montré que la transformation d'un processus d'affaires consiste à transformer chacune de ses vues en fonction de la question générique et la réponse spécifique obtenue. Nous avons alors développé une règle de transformation, pour chaque vue, par question générique et par réponse spécifique. Mais, *est-ce qu'on peut définir un ordre d'application des règles de transformation des vues indépendamment de la réponse à la question générique pour une transformation unitaire ?*

Pour répondre à cette question, nous allons analyser la dépendance entre les vues et les conflits possibles qui peuvent résulter de l'application d'une transformation *maximale*. Nous allons montrer qu'il est possible de définir un ordre d'exécution de cette transformation pour les vues. Par la suite, nous pouvons en déduire l'ordre d'exécution pour une transformation non maximale. Ici, nous traitons le cas d'une transformation unitaire. Les transformations globales associées à des questions différentes (voir sect. 6.2.1.3) seront abordées plus loin.

Rappelons que nous avons représenté un processus d'affaires par un quadruplet $\langle \text{REA}, D, O, I \rangle$ où *REA*, *D*, *O*, et *I* représentent les vues REA, dynamique, organisationnelle,

et informationnelle, respectivement. Nous noterons l'ensemble des éléments d'une vue v par Ev . Selon notre décomposition, nous obtenons la formule suivante :

$$Erea \cap Ed \cap Eo \cap Ei = \{\} \dots (f)$$

Notons $t_v : (v, q, r) \mapsto t_v(v, q, r)$, la règle de transformation associant une vue v à son image $v' = t_v(v, q, r)$ après une réponse r à une question générique q . Nous allons utiliser la notation abrégée $t_v(v)$ pour expliquer l'ordre d'exécution des règles de transformation indépendamment de la réponse à la question générique,

Nous utiliserons le symbole $<<$ pour définir l'ordre d'exécution des règles de transformation des vues. Par exemple $t_v(v1) << t_v(v2)$ signifie que la vue $v1$ doit être transformée avant la vue $v2$. Dans ce cas, l'ordre d'exécution des transformations sera noté par la séquence $< t_v(v1), t_v(v2) >$.

Finalement, nous utiliserons le symbole \perp pour signifier l'indépendance des règles de transformation entre les vues. Par exemple, $t_v(D) \perp t_v(I)$ signifie que la transformation de la vue dynamique est indépendante de la transformation de la vue informationnelle. Dans ce cas, l'ordre d'exécution des transformations $t_v(D)$ et $t_v(I)$ n'est pas important. Cet ordre d'exécution sera noté par l'ensemble $\{t_v(v1), t_v(v2)\}$.

À partir de la formule f , il n'y a pas de chevauchement entre les vues. Par contre, certaines vues font référence à d'autres. Voici les propriétés de dépendance entre les vues :

1. La vue dynamique dépend de la vue REA car elle définit les séquences des événements REA et les éléments de contrôle qui peuvent dépendre des ressources REA.
2. La vue organisationnelle dépend de la vue REA car elle fait référence aux événements REA réalisés par les agents.
3. La vue informationnelle dépend de la vue REA car elle définit la structure des ressources REA.
4. La vue dynamique dépend de la vue organisationnelle car elle définit les interactions entre les agents REA.
5. La vue informationnelle est indépendante des vues dynamique et organisationnelle.

À partir de ces informations, nous pouvons déduire que les transformations dépendent de la spécialisation de la vue REA. Plus précisément, elles dépendent du fait que la transformation ajoute ou supprime des éléments REA. Nous distinguons alors les cas suivants :

La transformation $t(rea)$ crée de nouveaux événements REA

Dans ce cas, il faut d'abord ajouter les éléments de la vue REA avant la transformation des autres vues. Car toutes les autres vues en dépendent (propriétés 1,2 et 3). Nous obtenons alors : $t_v(rea) \ll t_v(o)$, $t_v(rea) \ll t_v(d)$ et $t_v(rea) \ll t_v(i)$

À partir de la propriété 4, la vue organisationnelle doit être transformée avant la vue dynamique. Nous obtenons alors : $t(o) \ll t(d)$. Finalement, à partir de la propriété 5, nous obtenons la formule : $t_v(o) \perp t_v(i)$ et $t_v(d) \perp t_v(i)$.

À partir de cette analyse, l'ordre d'exécution des règles de transformation, dans le cas où la transformation crée de nouveaux événements, est défini par la séquence suivante :

$$T^{rea+} = \langle t_v(rea), \{t_v(o), t_v(i)\}, \{t_v(d), t_v(i)\} \rangle$$

La transformation $t(rea)$ supprime des événements REA:

Dans ce cas, il faut d'abord transformer les vues O, D et I avant de supprimer les éléments de la vue REA. En effet, pour la vue dynamique, il faut déterminer les prédécesseurs et les successeurs de chaque événement afin d'ajuster la séquence des événements. Pour la vue organisationnelle, il faut enlever les références aux événements avant de les supprimer de la vue REA. Finalement, pour la vue informationnelle, il faut supprimer les ressources créées ou utilisées exclusivement par les événements à supprimer. Nous obtenons alors :

$$t_v(o) \ll t_v(rea), t_v(d) \ll t_v(rea) \text{ et } t_v(i) \ll t_v(rea).$$

À partir de la propriété 4, la vue organisationnelle doit être transformée avant la vue dynamique. Nous obtenons alors : $T(o) \ll T(d)$. Finalement, à partir de la propriété 5, nous obtenons la formule : $t_v(o) \perp t_v(i)$ et $t_v(d) \perp t_v(i)$.

À partir de cette analyse, l'ordre d'exécution des transformations dans le cas où la transformation supprime des événements est défini par la séquence suivante :

$$T^{rea-} = \langle \{t_v(o), t_v(i)\}, \{t_v(d), t_v(i)\}, t_v(rea) \rangle$$

La transformation $t(rea)$ crée de nouveaux événements et en supprime d'autres

Dans ce cas, nous commençons d'abord par ajouter les événements REA. Nous appliquons les transformations selon la séquence T^{rea+} . Par la suite, nous appliquons les transformations selon la séquence T^{rea-} . Nous obtenons alors la séquence $T^{rea+} \theta T^{rea-}$ où le symbole θ désigne la concaténation de séquences.

La transformation ne modifie pas la vue REA

Ce cas est un exemple de transformation non maximale. Quand la vue REA ne change pas, la vue informationnelle ne change pas non plus. Selon la quatrième propriété de dépendance, la vue organisationnelle doit être transformée avant la vue dynamique. Nous obtenons alors : $t_v(o) \ll t_v(d)$. L'ordre d'exécution des transformations est défini, dans ce cas, par la séquence $\langle t_v(o), t_v(d) \rangle$.

Nous avons démontré, ci-dessus, qu'il est possible de déterminer un ordre d'exécution des règles de transformation entre les vues d'un processus d'affaires pour une réponse à une question. Toutefois, nous avons été confrontés à certains problèmes bien connus aux approches transformationnelles lors de l'application de règles de transformation concernant plusieurs questions. En effet, les instances des modèles des vues peuvent se chevaucher dans ce cas. Par conséquent, l'ordre d'exécution des règles de transformation peut poser des problèmes que nous discutons dans la section 6.4.

6.2.2 Exemple : Spécialisation du processus de vente et distribution

Pour montrer notre approche transformationnelle, nous allons reprendre le processus de vente et distribution présenté dans le chapitre 5. Nous rappelons que le processus de vente est un processus d'échange qui consiste à transférer le droit de propriété d'un produit à un acheteur en échange d'une somme d'argent. La distribution est un processus de conversion qui change l'attribut emplacement du produit. La Figure 6.13 montre un diagramme BPMN du processus de vente et distribution.

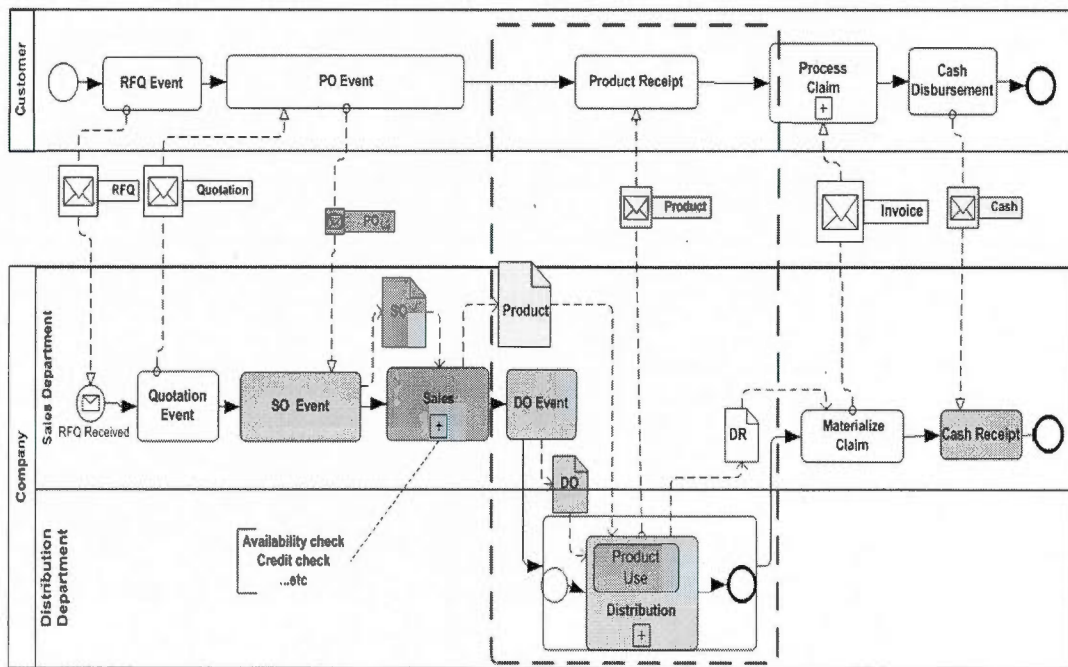


Figure 6.13 Le processus de vente et distribution.

Dans ce qui suit, nous allons spécialiser ce processus en appliquant deux questions. La première question (*q3*) traite le sujet de sous-traitance du processus de distribution (fig. 6.14). La deuxième question (*q6*) traite du calcul automatique des créances (fig. 6.15). La représentation de la question *q6* montre que celle-ci n'est valide que pour les processus d'échange. En effet, le concept de créance n'existe pas pour les processus de conversion.

Nous cherchons à obtenir un nouveau processus d'affaires de vente et distribution après une réponse positive (vraie) aux questions *q3* et *q6*. En d'autres termes, nous visons un processus de vente et distribution dont la distribution est sous-traitée à un partenaire et dont la gestion des créances repose sur le principe du *Two-way-match* (sect. 5.1). Dans la suite de cette section, nous désignons par '1' une réponse positive, et par '0' une réponse négative dans le cas d'une réponse binaire (i.e. oui ou non). De là, le chemin de spécialisation de ce nouveau processus à partir du processus initial est $\langle (Distribution, q3, 1), (q6, 1) \rangle$.

```

<!-- ***** -->
<!-- Sub-contracting REA pattern -->
<!-- ***** -->
<question id="q3">
  <name>SUBCONTRACT_NESTED_PROCESS</name>
  <description>
    Some business process may be outsourced to external agent.
  </description>
  <core answerType="AnswerType.BOOLEAN" >
    Does the organization plan outsourcing the business process {0} to a partner?
  </core>
  <parameters>
    <parameter>Process</parameter>
  </parameters>
</question>

```

Figure 6.14 La représentation de la question de sous-traitance.

```

<!-- ***** -->
<!-- Materialized Claim REA pattern -->
<!-- ***** -->
<question id="q6">
  <name>CLAIM_COMPUTED_AUTOMATICALLY</name>
  <description>
    By default, business processes use the Three-way match strategy for
    claim settlement. In some cases, business processes use the Two-way
    match strategy.
  </description>
  <core answerType="AnswerType.BOOLEAN" >
    Does the business process {0} support a Two-way match strategy for claim processing?
  </core>
  <parameters>
    <parameter>ExchangeProcess</parameter>
  </parameters>
</question>

```

Figure 6.15 La représentation de la question du calcul automatique des créances.

Notre approche de spécialisation ne préconise pas la sauvegarde des modèles intermédiaires obtenus par la transformation du processus après chaque transformation (i.e. après chaque question/réponse). Dans cet exemple, nous allons expliciter les vues après l'application de chacune des transformations à titre explicatif seulement. Nous allons aussi montrer la chaîne de valeur pour expliquer la transformation.

Notons, $t_u : (p, q, r) \mapsto t_u(p, q, r)$, la transformation associant un processus d'affaire p à un nouveau processus d'affaires $p' = t_u(p, q, r)$ après une réponse r à une question générique q . Nous représentons alors la transformation qui consiste à sous-traiter le processus de distribution par $t_u(\text{Distribution}, q3, 1)$ et la transformation permettant le calcul automatique de la valeur des créances par $t_u(\text{Vente et Distribution}, q6, 1)$.

Après l'application de la transformation $t_u(\text{Distribution}, q3, 1)$, un nouveau processus d'échange s'ajoute à la chaîne de valeur selon le patron de sous-traitance présenté dans la section 6.1. Ce nouveau processus consiste en l'achat du service de distribution (*Distribution Service Purchase*). L'événement d'incrément est l'acquisition du service de distribution. L'événement de décrément est le paiement du service acheté. L'application de la transformation $t_u(\text{Vente et Distribution}, q6, 1)$ ne change pas la chaîne de valeur. La Figure 6.16 montre la chaîne de valeur qui tient compte de la sous-traitance de la distribution.

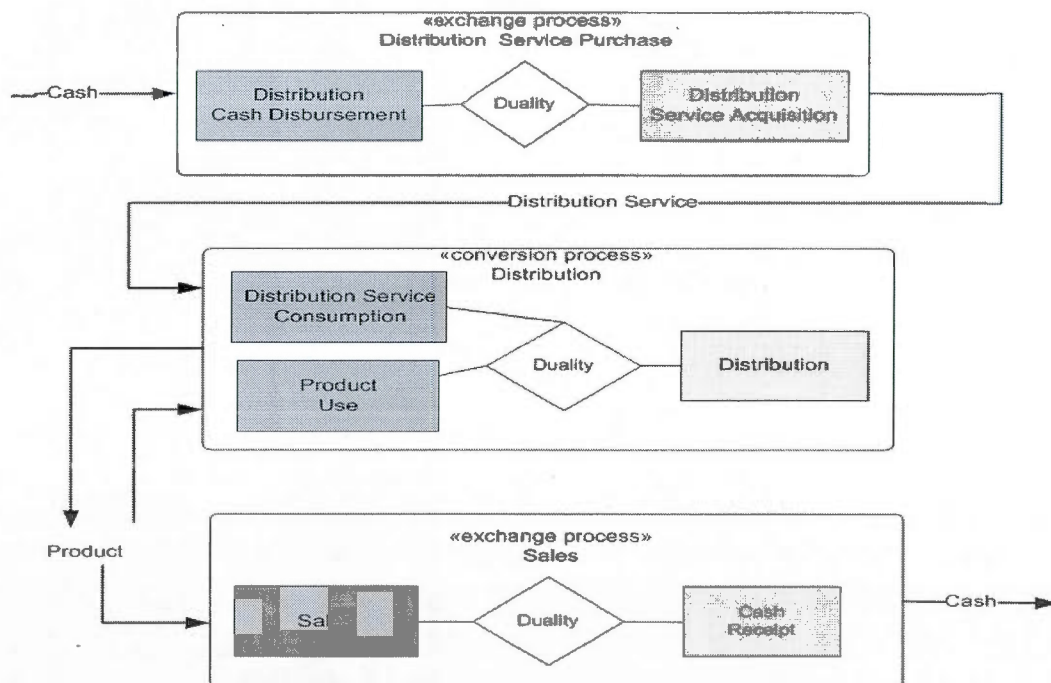


Figure 6.16 La nouvelle chaîne de valeur après la sous-traitance.

6.2.2.1 Transformation de la vue REA

Dans le chapitre 5 (sect. 5.1), nous avons montré la vue REA du processus de vente et de distribution. Dans cette section, nous montrons les variations que subit cette vue après l'application des transformations $t_u(\text{Distribution}, q3, 1)$ et $t_u(\text{Vente et Distribution}, q6, 1)$, dans cet ordre.

La transformation $t_u(\text{Distribution}, q3, 1)$ crée un nouveau processus d'échange qui consiste en l'achat du service de distribution. La Figure 6.17 montre la vue REA de base du processus d'achat du service de distribution selon le patron de sous-traitance. L'acquisition du service de distribution (*Distribution Service Acquisition*) représente l'événement d'incrément. Le paiement pour le service acheté (*Distribution Cash Disbursement*) représente l'événement de décrétement. Le service de distribution (*Distribution Service*) acheté du fournisseur (*Distribution Provider*) est une ressource transitoire qui est consommée aussitôt créée. D'après la chaîne de valeur de la Figure 6.16, le service de distribution sera consommé par le processus de distribution. La Figure 6.18 montre la vue complète du processus d'achat du service de distribution (fig. 6.17) en tenant compte des événements du traitement de la créance (i.e. la vue REA augmentée par les événements d'affaires et informationnels).

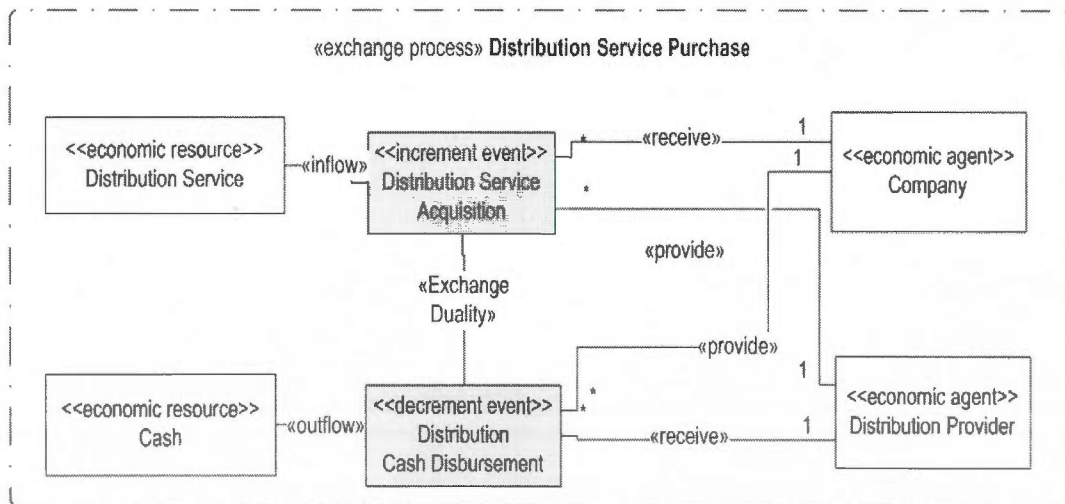


Figure 6.17 La vue REA de base du processus d'achat du service de distribution.

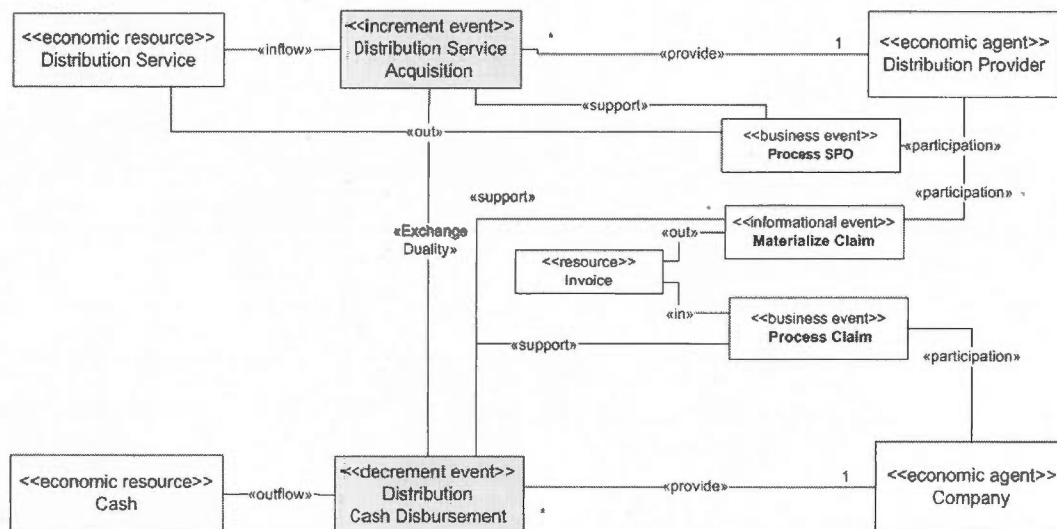


Figure 6.18 La vue REA complète du processus d'achat du service de distribution.

Selon le patron d'affaires de sous-traitance, le processus de distribution subit un changement. En effet, le service de distribution acheté est une ressource transitoire. Elle sera consommée dans le processus de distribution. Le nouvel événement de consommation du service (i.e. événement de décrétement) sera jumelé à l'événement d'utilisation du produit (i.e. l'autre événement de décrétement) pour réaliser la conversion via la relation de dualité avec l'événement de distribution (i.e. événement d'incrément). La Figure 6.19 montre le nouveau modèle REA du processus de distribution. Nous ne montrons que la vue de base pour simplifier la lecture du modèle.

À titre indicatif, la Figure 2.20 montre le modèle REA global du processus de vente et distribution intégrant la vente (fig. 5.6), l'achat de service de distribution (fig. 6.18), et la distribution (fig. 6.19). Le modèle REA global peut être obtenu en intégrant les différentes vues REA (*views integration*) des processus de la chaîne de valeur tel que proposé par Davids (1997). Cette étape de recomposition fait partie de notre méthodologie de spécification de processus d'affaires décrite dans le chapitre 3. Nous y reviendrons à la conclusion.

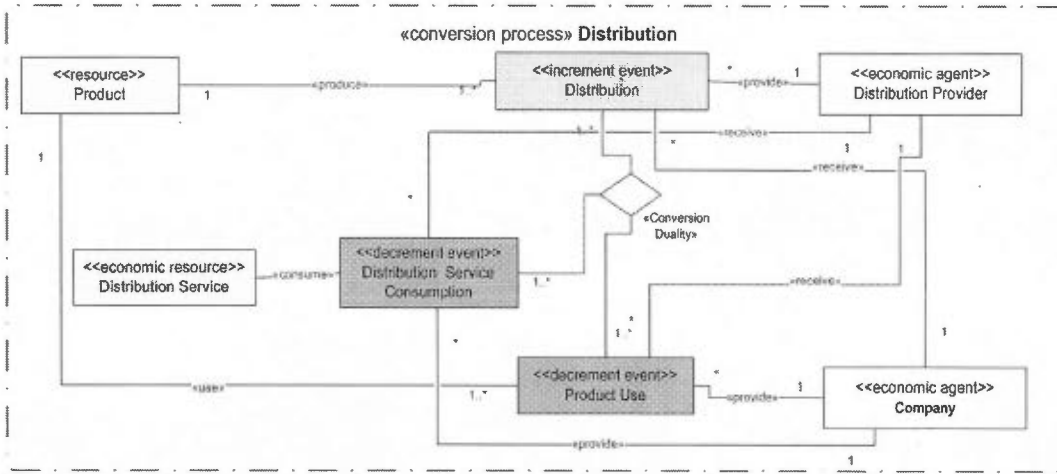


Figure 6.19 Le processus de distribution après la sous-traitance.

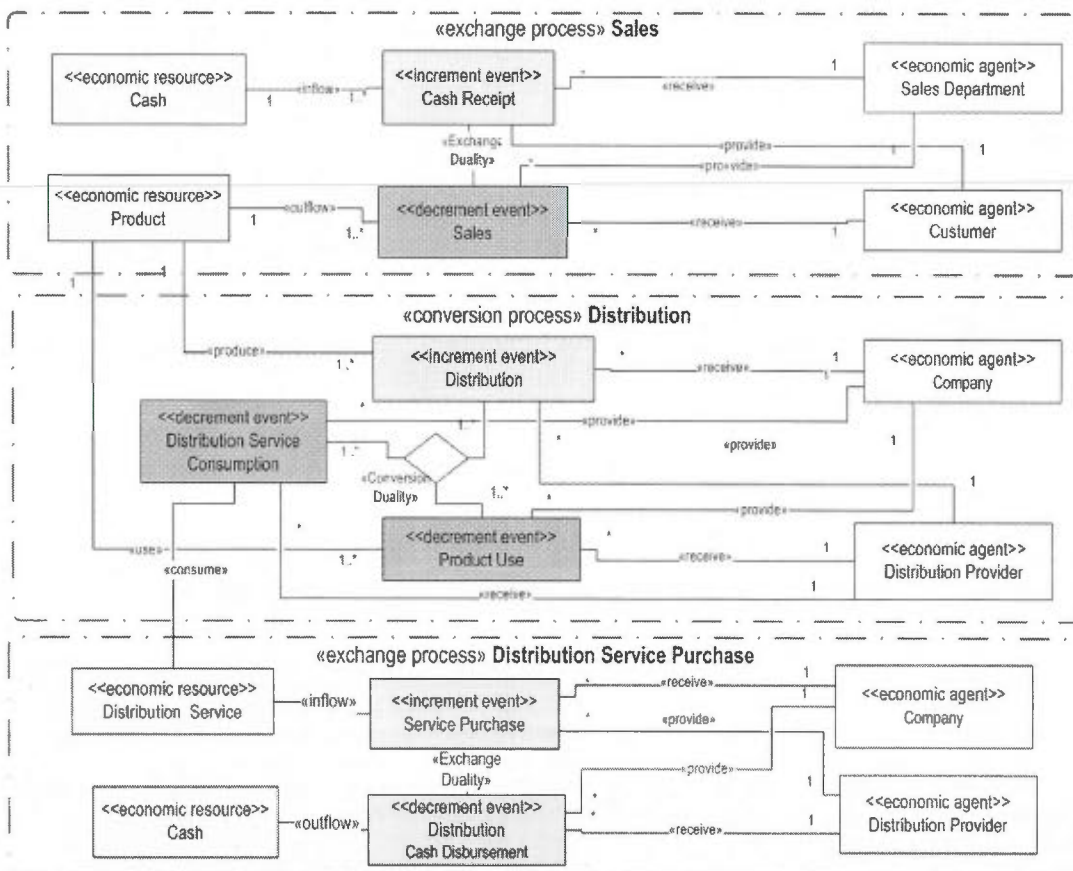


Figure 6.20 Le modèle REA après la sous-traitance de la distribution.

Considérons maintenant l'application de la transformation $t_u(\text{Vente et Distribution}, q6, 1)$. Rappelons que cette question traite du calcul et de la vérification des créances pour les processus d'échanges. Une réponse positive indique que la valeur de la créance peut être calculée automatiquement à partir des engagements (*commitments*). Dans ce cas, les processus utilisent la stratégie de vérification *Two-way-match* (sect. 6.1). Dès lors, l'agent économique qui doit créer et envoyer la facture pour chacun des processus d'échanges (i.e. la vente et l'achat du service de distribution) n'est pas tenu de le faire. En effet, l'agent partenaire qui devait recevoir la facture peut alors compléter l'échange en calculant automatiquement la valeur de la créance (i.e. *unbalancedValue* de l'entité *MaterializedClaim* du patron d'affaires). À partir de là, l'événement informationnel de matérialisation et d'envoi de la créance sera supprimé de la vue REA pour les deux processus d'échanges. La vue REA du processus de distribution ne subira aucun changement car le concept de créance ne s'applique pas aux processus de conversion. Les Figures 6.21 et 6.22 montrent, respectivement, la vue REA du processus de vente et la vue REA du processus d'achat du service de distribution après l'application des transformations $t_u(\text{Distribution}, q3, 1)$ et $t_u(\text{Vente et Distribution}, q6, 1)$, dans cet ordre.

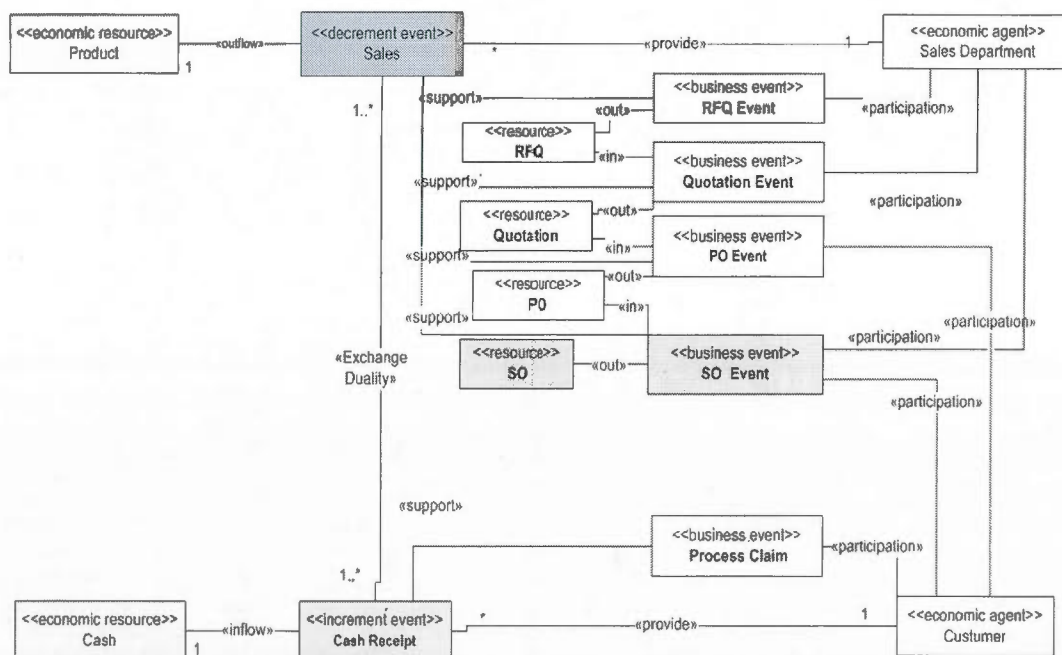


Figure 6.21 La vue REA finale du processus de la vente.

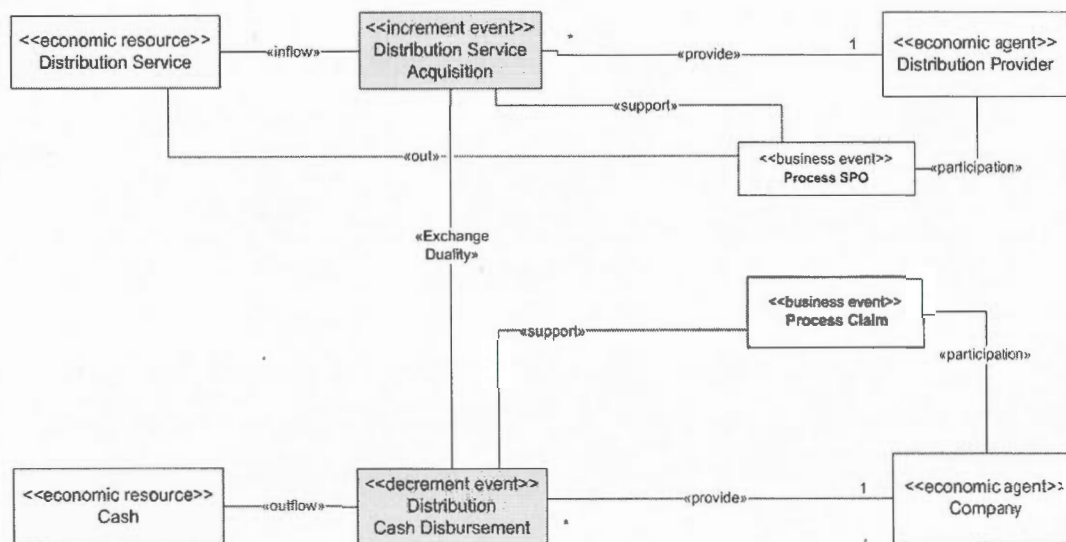


Figure 6.22 La vue REA finale du processus d'achat du service de distribution.

6.2.2.2 Transformation de la vue organisationnelle

La Figure 5.11 du chapitre 5 montre la vue organisationnelle du processus de vente et distribution. Pour simplifier l'interprétation de cette vue, nous utiliserons les concepts graphiques de la norme BPMN. Les éléments qui ne font pas partie de la vue organisationnelle, mais qui y sont référencés, sont représentés en lignes pointillées.

Après l'application de la transformation $t_u(\text{Distribution}, q3, 1)$, le processus de distribution est confié à un agent externe. Par conséquent, la vue organisationnelle du processus de vente et distribution subit des changements du fait que les tâches du processus sous-traité sont exécutées par un agent externe à l'organisation. De plus, un nouveau processus d'achat de service de distribution est ajouté à la chaîne de valeur. Les événements du processus de distribution seront exécutés par un agent tiers (*Distribution Provider*). C'est ce dernier qui fournit le service de distribution du nouveau processus d'échange (i.e. processus d'achat du service de distribution). Pour ce nouveau processus, l'agent interne recevra l'événement d'achat du service de distribution (i.e. l'événement d'incrément) et fournira l'événement de paiement (l'événement de décrétement).

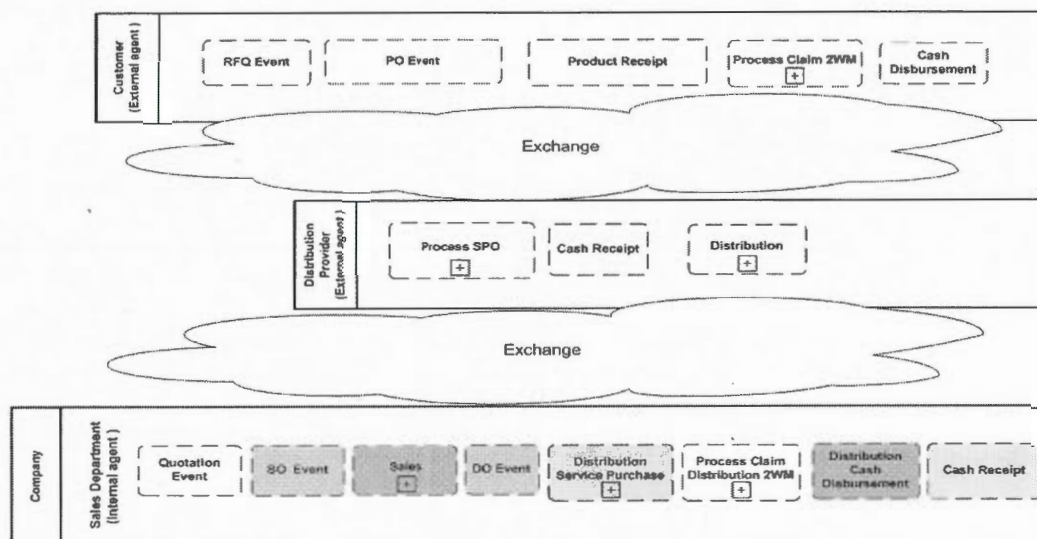


Figure 6.23 La vue organisationnelle finale du processus de vente et distribution.

Nous avons vu que l'application de la transformation $t_u(\text{Vente et Distribution}, q6,1)$ a modifié la vue REA des deux processus d'échange (fig. 6.21 et 6.22). En effet, l'événement de matérialisation et d'envoi de la créance (*Materialize Claim*) est supprimé. Par conséquent, la référence à cet événement est supprimée de la vue organisationnelle pour les deux processus d'échanges. La Figure 6.23 montre la vue organisationnelle finale du processus de vente et distribution. Les événements montrés font référence à la vue REA. L'événement *Process Claim 2WM* indique que l'événement d'affaires qui établit le paiement se base sur la stratégie de vérification *Two-way-match*.

6.2.2.3 Transformation de la vue dynamique

La Figure 5.12 du chapitre 5 montre la vue dynamique du processus de vente et distribution. Comme pour la vue organisationnelle, nous utiliserons les concepts graphiques de la norme BPMN pour l'interprétation de cette vue. Les éléments qui ne font pas partie de la vue dynamique sont représentés en lignes pointillées.

La transformation de la vue dynamique dépend de la transformation de la vue REA et de la transformation de la vue organisationnelle (sect. 6.1). Notre approche de transformation de la vue dynamique supporte trois opérations de spécialisation : l'ajout, la suppression et le

déplacement. Ces opérations spécialisent les liens comportementaux (e.g. *BPMN connectors*), les éléments de contrôle qui représentent la façon dont les événements sont connectés (e.g. *BPMN gateways*) et les événements de contrôles (e.g. *BPMN events*).

Après l'application de la transformation $t_u(\text{Distribution}, q3, 1)$, un nouveau processus d'achat de service de distribution est ajouté. Comme le processus de distribution n'est plus effectué à l'interne mais par un agent externe à l'organisation, les flux de séquences entre les événements du processus de vente et celui de la distribution deviennent des messages (e.g. la séquence d'envoi du contrat de distribution). En général, la transformation de la vue dynamique consiste à ajuster les successions, les messages et les éléments de contrôle suite à l'ajout, la suppression et au déplacement des événements REA.

Après l'application de la transformation $t_u(\text{Vente et Distribution}, q6, 1)$ aux deux processus d'échange, les éléments comportementaux liés à l'événement de matérialisation et d'envoi de la créance seront ajustés. Un exemple d'ajustement serait de court-circuiter l'événement de matérialisation de la créance (i.e. relier la séquence qui précède l'événement de matérialisation de la créance à l'événement qui le succède). Finalement, le message d'envoi de la facture sera supprimé. La Figure 6.24 montre la vue dynamique finale du processus de vente et de distribution. Les éléments de la vue dynamique sont illustrés en gras foncé. Les événements montrés font référence à la vue REA.

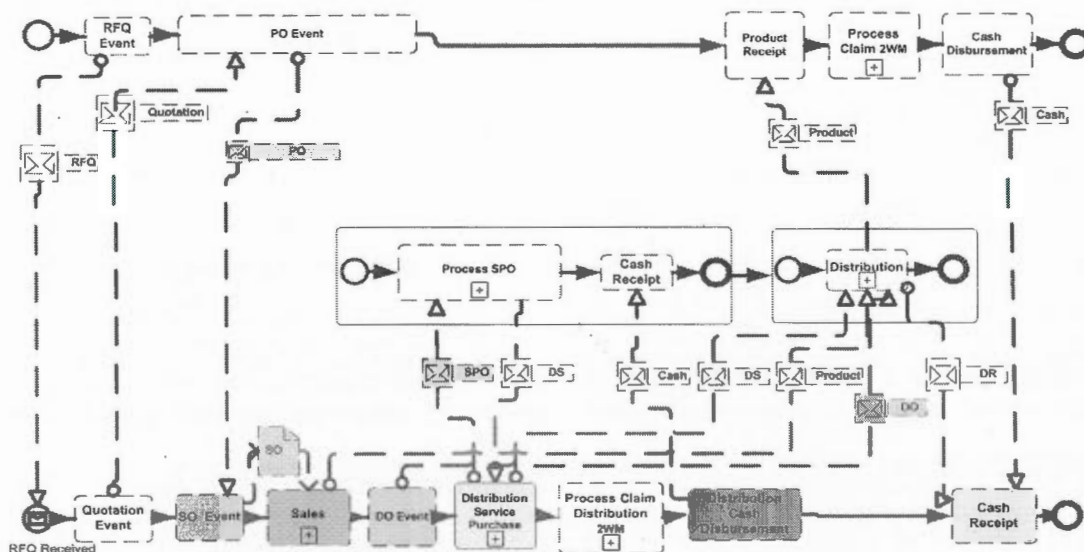


Figure 6.24 La vue dynamique finale du processus de vente et distribution.

6.2.2.4 Transformation de la vue informationnelle

La vue informationnelle représente les ressources manipulées dans un processus d'affaires. Comme illustré dans le chapitre 5 (fig. 5.13 et fig. 5.14), nous utilisons le diagramme de classe du langage UML pour la représentation de la vue informationnelle. La spécialisation de la vue informationnelle consiste en un ensemble de transformations de modèles bien connues dans le domaine de l'orienté-objet.

Suite à l'application de la transformation $t_u(Distribution, q3, 1)$, un nouveau processus d'achat de service de distribution est ajouté à la chaîne de valeur. La création de la vue informationnelle de ce nouveau processus d'affaires suit la même technique décrite dans le chapitre 5 (sect. 5.1.6). Nous allons l'appliquer pour la modélisation de la vue informationnelle du nouveau processus. Pour bien situer ce processus, nous allons utiliser le diagramme BPMN de la Figure 6.25 qui montre le processus de vente et distribution après l'ajout du nouveau processus (i.e. après l'application de la transformation $t_u(Distribution, q3, 1)$). Le cadre en lignes pointillées montre le nouveau processus d'achat du service de distribution.

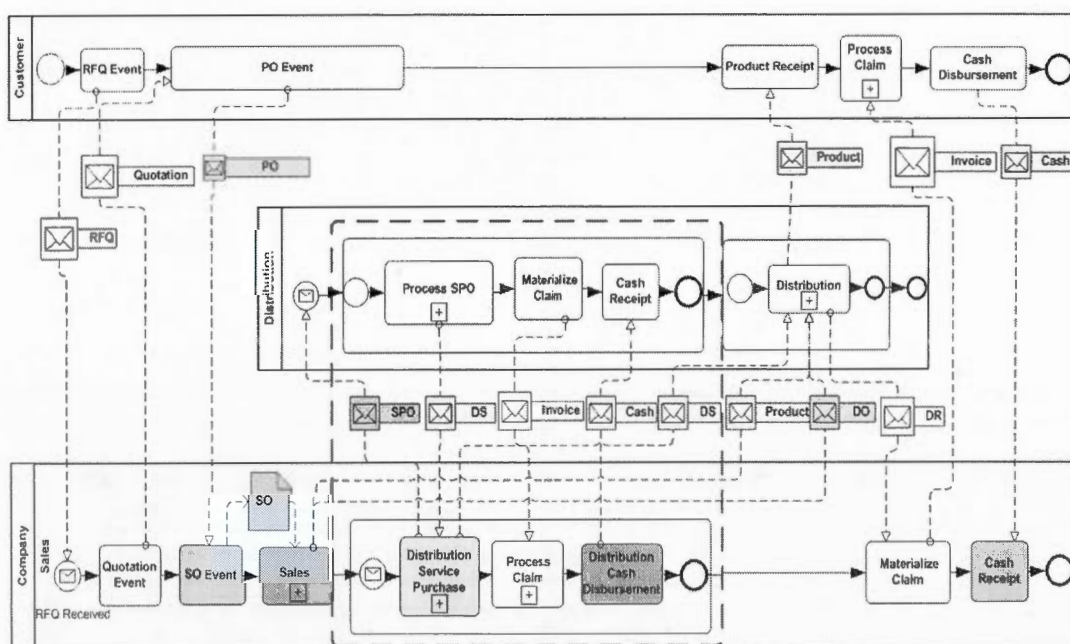


Figure 6.25 Le processus de vente et distribution après la sous-traitance de la distribution.

Pour modéliser la vue informationnelle, nous commençons par le contrat d'achat de service *SPO* (*Service Purchase Order*). Un contrat contient des clauses qui représentent les obligations d'incrément et de décrétement. Ces obligations forment les lignes du contrat. Dans notre cas, le contrat d'achat contient des lignes d'achat et des lignes de paiement. Les lignes d'achat sont des obligations d'incrément désignant les items à acheter. Les lignes de paiement sont des obligations de décrétement désignant les paiements à effectuer pour les items à acheter. Les lignes de paiement (i.e. les obligations de décrétement) et les items à acheter (i.e. les obligations d'incrément) sont reliés par la relation de réciprocité. Finalement, les obligations sont reliées aux ressources économiques correspondantes par la relation de réservation. Le processus de modélisation se poursuit par l'ajout de l'entité qui représente la créance. Dans notre cas, la créance REA est matérialisée en la facture qui sera envoyée par le fournisseur du service de distribution. La créance est reliée aux obligations (i.e. les lignes de paiement et les items à acheter) par les relations de *materialization* et de *settlement* selon le patron d'affaires de matérialisation de la créance (sect. 6.1).

La sous-traitance est un contrat par définition. Dès lors, il faut ajouter un accord qui va réglementer les contrats individuels d'achat de service de distribution selon le patron d'affaires REA du contrat (sect. 6.1). Par conséquent, il faut ajouter une nouvelle classe *Agreement* et des associations bidirectionnelles la reliant aux agents économiques et à l'entité qui représente le contrat REA (*Service Purchase Order*). La Figure 6.26 montre la vue informationnelle du nouveau processus d'achat du service de distribution.

Après l'application de la transformation $t_n(\text{Vente et Distribution}, q6,1)$ au processus d'achat du service de distribution, la matérialisation de la créance n'est plus nécessaire. Par conséquent, l'entité *MaterializedClaim* sera supprimée de la vue informationnelle. Avant de supprimer cette entité, il faut remonter les relations de *materialization* et de *settlement* au niveau de la créance pour conserver un modèle informationnel valide. La Figure 6.27 montre la vue informationnelle finale du processus d'achat du service de distribution.

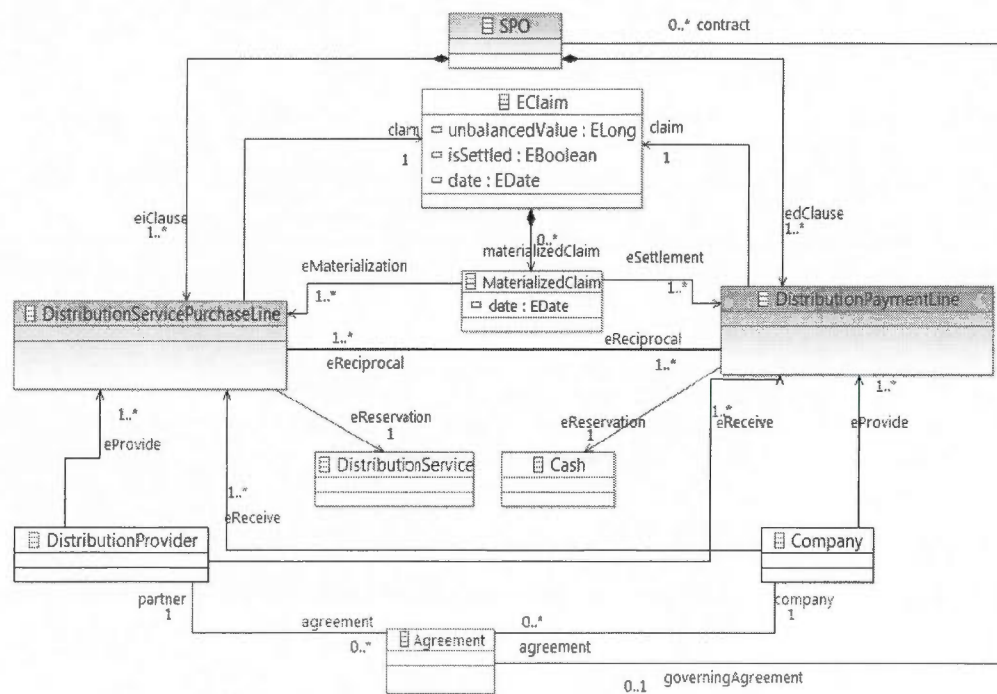


Figure 6.26 La vue informationnelle du processus d'achat du service de distribution.

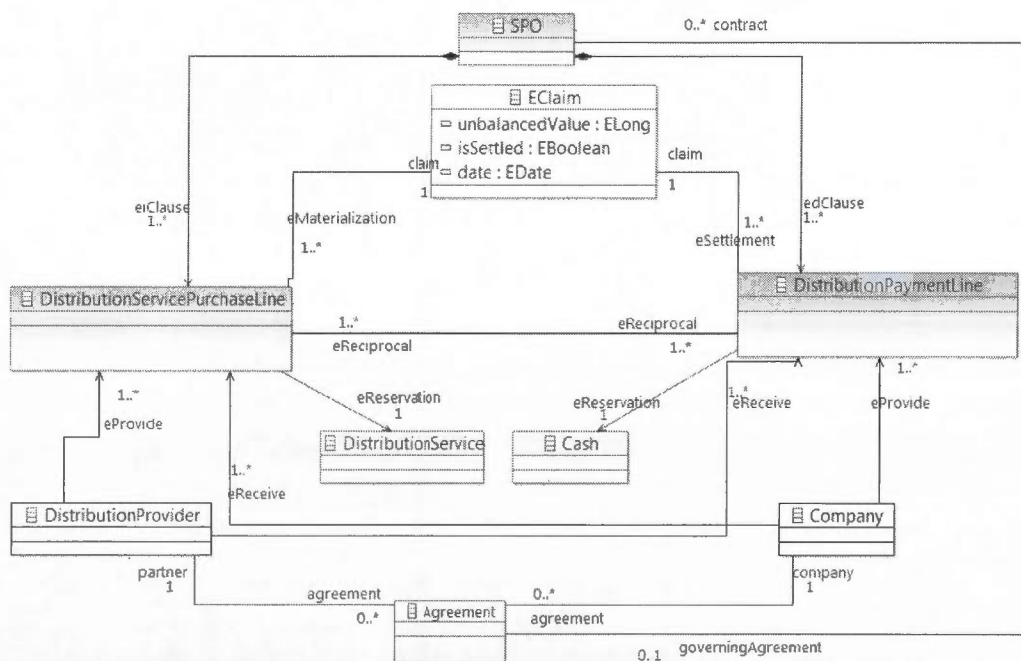


Figure 6.27 La vue informationnelle finale du processus d'achat du service de distribution.

Nous appliquons ces mêmes transformations (i.e. $t_u(\text{Distribution}, q3, 1)$ et $t_u(\text{Vente et Distribution}, q6, 1)$) à la vue informationnelle du processus de vente. Selon le patron de soustraction, la vue informationnelle du processus de vente ne doit pas être affectée par la transformation $t_u(\text{Distribution}, q3, 1)$. Après l'application de la transformation $t_u(\text{Vente et Distribution}, q6, 1)$ à ce processus, la matérialisation de la créance n'est plus nécessaire. Nous appliquons alors les mêmes transformations que celles appliquées à la vue informationnelle du processus d'achat du service de distribution. La Figure 6.28 montre la vue informationnelle finale du processus de vente après l'application des transformations $t_u(\text{Distribution}, q3, 1)$ et $t_u(\text{Vente et Distribution}, q6, 1)$, dans cet ordre.

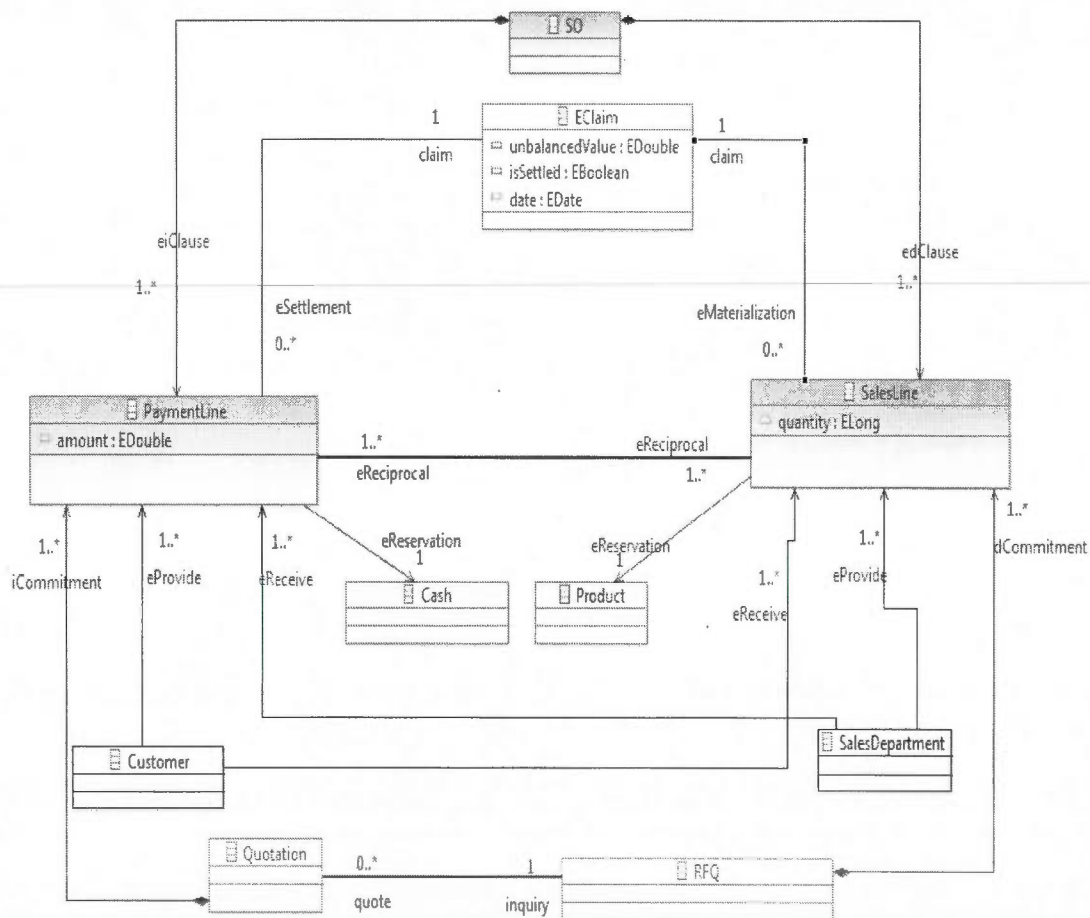


Figure 6.28 La vue informationnelle finale du processus de vente.

Finalement, la dernière vue informationnelle à transformer est celle du processus de distribution. Selon le patron de sous-traitance, la nouvelle ressource économique *Distribution Service* sera consommée dans le processus de distribution. L'événement de consommation du service (*Distribution Service Consumption*) sera jumelé à l'événement d'utilisation du produit (i.e. l'événement de décrétement) pour réaliser la conversion via la relation de dualité avec l'événement de distribution (i.e. l'événement d'incrément).

Le processus de distribution est un processus de conversion. Il n'est donc pas concerné par la transformation $t_u(\text{Vente et Distribution}, q6, 1)$. La Figure 6.29 montre la vue informationnelle finale du processus de distribution après l'application des transformations $t_u(\text{Distribution}, q3, 1)$ et $t_u(\text{Vente et Distribution}, q6, 1)$, dans cet ordre.

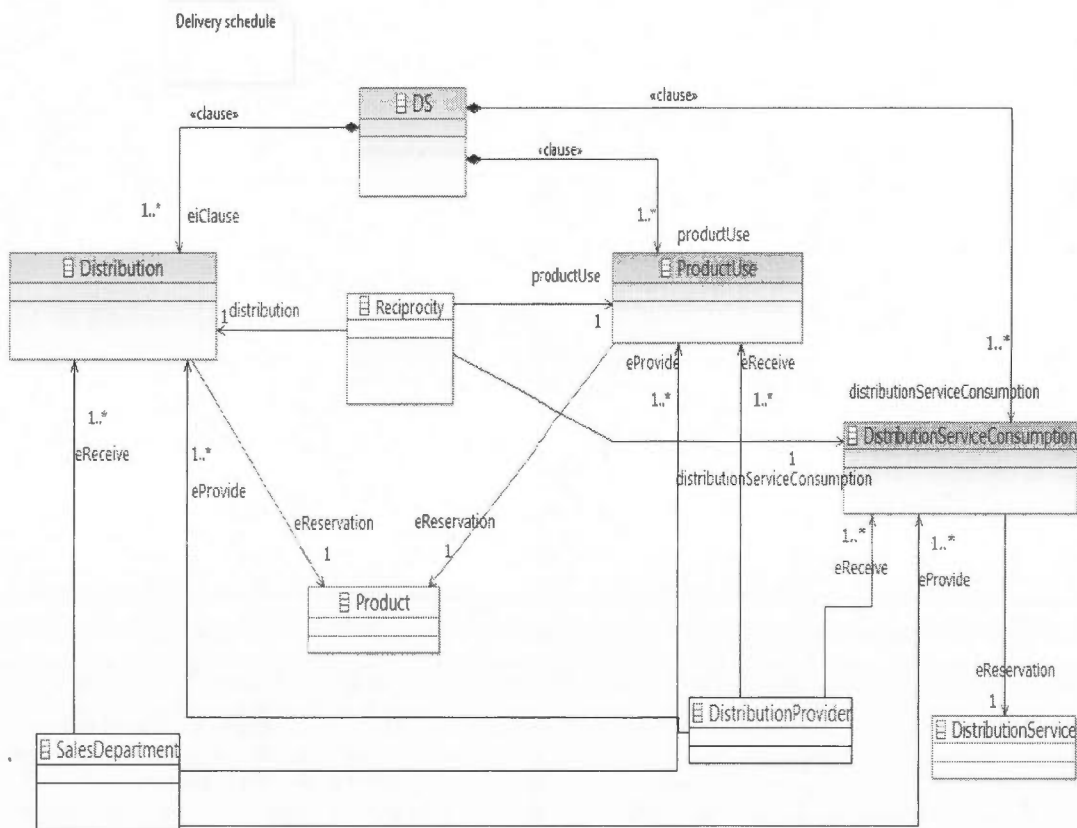


Figure 6.29 La vue informationnelle finale du processus de distribution.

Finalement, le nouveau processus de vente et distribution, obtenu après la spécialisation du processus initial (fig 6.13) par les transformations $t_u(\text{Distribution}, q3, 1)$ et $t_u(\text{Vente et Distribution}, q6, 1)$ respectivement, est montré avec la notation BPMN dans la Figure 6.30. Le chemin de spécialisation de ce nouveau processus générique à partir du processus initial est $\langle (\text{Distribution}, q3, 1), (q6, 1) \rangle$. Comme nous pouvons à nouveau le constater, ces transformations ont peu à voir avec la spécialisation en orienté-objet.

Dans la section qui suit, nous allons décrire notre implémentation des transformations présentées dans cette section. Nous montrerons aussi comment implémenter les séquences d'exécution des règles de transformation décrites dans la section 6.2.1.

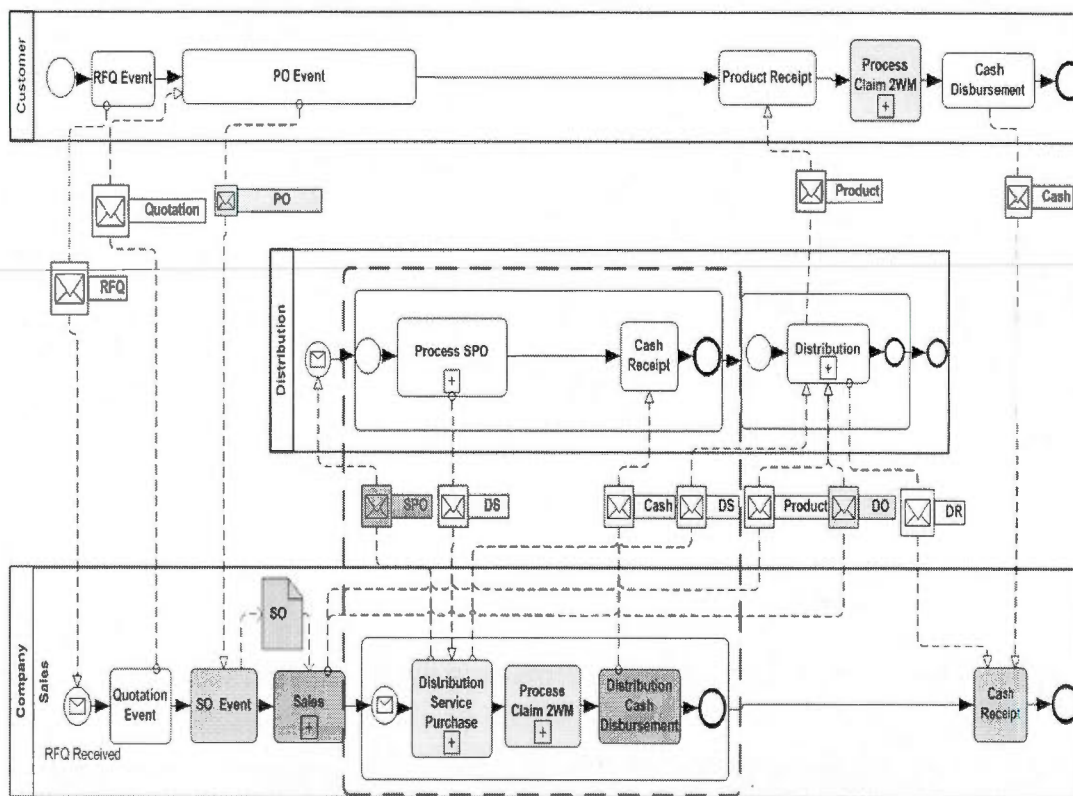


Figure 6.30 Le nouveau processus d'affaires de vente et distribution.

6.3 Implémentation des règles de transformation

Nous avons implémenté la spécialisation avec le système de gestion de règles d'affaires Drools (JBoss, 2012). Dans cette section, nous présentons d'abord JBoss Drools. Nous illustrons par la suite notre implémentation des règles avec des exemples.

6.3.1 JBoss Drools pour la représentation des règles

Drools est un système de gestion de règles d'affaires (*Business Rules Management System*) qui utilise un moteur d'inférence à chaînage avant. Drools est basé sur l'algorithme *Rete* de filtrage par motif (*Pattern matching*). Actuellement, les règles peuvent être écrites en Java, MVEL (*MVFLEX Expression Language*), Python, et Groovy. Pour des règles plus claires et compréhensibles par les analystes d'affaires, Drools permet aussi de spécifier des règles en langage spécifique aux domaines (*Domain Specific Language*). De plus, Drools offre un éditeur graphique pour guider la création de règles et de tables de décisions.

Drools est composé de cinq projets : (i) *Drools Guvnor* pour la gestion des règles d'affaires, (ii) le moteur de règles *Drools Expert*, (iii) la suite de gestion de processus d'affaires *jBPM* (précédemment *Drools flow*), (iv) *Drools Fusion* pour le traitement des événements complexes (*Complex Event Processing*), et (v) *Drools Planner* pour l'automatisation de planifications. Ce dernier supporte aussi la planification de problèmes *NP difficile* en utilisant des algorithmes de méta-heuristique tels que *tabu search* et *simulated annealing* (JBoss, 2012).

Une règle Drools est composée d'une partie condition et d'une partie action. La forme d'une règle Drools est : `when {condition} then {action}`. Les règles Drools sont définies dans des fichiers *drl*. Comme les autres moteurs de règles basés sur l'algorithme *Rete*, Drools analyse les objets présents dans sa mémoire de travail ainsi que les conditions de l'ensemble des règles (*ruleset*) pour déterminer, parmi les instances des règles, celles qui doivent être ajoutées à son agenda d'exécution.

La Figure 6.31 illustre un exemple montrant la règle de transformation de la vue informationnelle dans le cas d'une réponse positive à la question du contrat (i.e. *Est-ce que le déroulement du processus {0} est régi par un accord ?*). Cette règle est écrite en Java sous le

fichier *InformationalView.drl*. Comme on peut le constater, cette règle de transformation est générique. À la ligne 2, nous définissons le groupe de flux de règles auquel appartient cette règle. Nous expliquons ce concept dans la section 6.3.2 lorsque nous traiterons l'ordre d'exécution des transformations des vues. La partie *when* récupère l'instance de la question spécifique depuis la mémoire de travail dans la variable *\$q* (à la ligne 5). *\$q.param(0)* contient le processus spécifique de la chaîne de valeur pour lequel on veut ajouter l'accord économique. Ce processus est une instance du paramètre générique de la question du contrat (fig. 6.10).

La partie *then* de cette règle procède comme suit :

1. On récupère les parties du contrat REA (lignes 9, 10 et 11).
2. On ajoute une nouvelle classe *Agreement* et ses attributs *dateFrom* et *dateTo* à la vue informationnelle (lignes 13 à 17).
3. On ajoute des associations bidirectionnelles entre cette nouvelle classe et les agents économiques selon le patron d'affaires du contrat (lignes 19 à 25). La signature de la méthode *addEReference* est :

```
public static EReference addEReference (Resource view,
                                       String eClassFrom,
                                       String eClassTo,
                                       String eReferenceName,
                                       int lowerBound,
                                       int upperBound,
                                       EReference opposite,
                                       boolean isContainment)
```

où *view* désigne la vue informationnelle. *eClassFrom* et *eClassTo* représentent les extrémités de la référence. *eClassTo* est le type de la référence (*eReferenceType*). *lowerBound* et *upperBound* sont les cardinalités minimales et maximales, respectivement. Le paramètre *opposite* désigne la référence opposée. Finalement, *isContainment* indique si la référence est de type agrégation (i.e. le paramètre *isContainment* est égal à *true*).

4. On ajoute les associations bidirectionnelles entre cette nouvelle classe et l'entité qui représente le contrat REA selon le patron d'affaires du contrat (lignes 28 à 31).
5. On supprime, de la vue informationnelle, tous les éléments liés exclusivement aux événements d'identification de l'agent partenaire (ligne 34).

La Figure 6.32 montre la règle de transformation de la vue REA dans le cas d'une réponse positive à la question du contrat. Comme pour la règle de transformation de la vue informationnelle, la partie *when* récupère l'instance de la question spécifique depuis la mémoire de travail dans la variable *\$q*. La partie action supprime les événements d'affaires REA liés à l'identification de l'agent partenaire. Les ressources REA liées exclusivement à ces événements seront aussi retirées de la vue.

```

1  rule "AGREEMENT IN THE PROCESS-INFORMATIONAL VIEW"
2      ruleflow-group "Informational_E-"
3
4      when
5          $q : Question(answer.value="true",
6              genericQuestion.name=GQuestions.AGREEMENT_IN_THE_PROCESS);
7      then
8          # Identify REA economic agents
9          Message m = EMFDynamicViewHandler.getREAContractMessage(bProcess, $q.param());
10         Agent agent1 = (Agent) m.getSourcePart();
11         Agent agent2 = (Agent) m.getTargetPart();
12         # 1. Add agreement structure according to the REA contract business pattern
13         EClass clazz = EMFInfViewHandler.addEClass(view,Globals.AGREEMENT_CLASS,
14             BpPackage.eINSTANCE.EResource());
15         # Add Attributes
16         EMFInfViewHandler.addEAttribute(clazz, "dateFrom", EcorePackage.eINSTANCE.getEDate());
17         EMFInfViewHandler.addEAttribute(clazz, "dateTo", EcorePackage.eINSTANCE.getEDate());
18         # Add bidirectional references between the agreement class and agents classes
19         EReference ref1 = EMFInfViewHandler.addEReference(view, clazz, agent1,
20             agent1.toLowerCase(), 1, 1, null, false);
21         EMFInfViewHandler.addEReference(view, agent1, clazz,Globals.AGREEMENT_CLASS.toLowerCase(),
22             ,ETypedElement.UNBOUNDED_MULTPLICITY, ref1, false);
23         EReference ref2 = EMFInfViewHandler.addEReference(view, clazz, agent2,
24             agent2.toLowerCase(), 1, 1, null, false);
25         EMFInfViewHandler.addEReference(view, agent2, clazz,Globals.AGREEMENT_CLASS.toLowerCase(),
26             ,ETypedElement.UNBOUNDED_MULTPLICITY, ref2, false);
27         # Add bidirectional references between the agreement class and the contract class
28         EReference ref3 = EMFInfViewHandler.addEReference(view, clazz, m.getResource(),"contract",
29             0, ETypedElement.UNBOUNDED_MULTPLICITY,
30             null, false);
31         EMFInfViewHandler.addEReference(view, m.getResource, clazz,Globals.AGREEMENT_CLASS.toLowerCase(),
32             0,1, ref3, false);
33         # 2. Remove structure related to the removed REA events
34         EMFInfViewHandler.removeElementsByEventTypeName(bProcess, $q.param(), "REAAgentIdentification");
35     end

```

Figure 6.31 Exemple de règle de transformation de la vue informationnelle.

```

1 rule "AGREEMENT IN THE PROCESS-REA VIEW"
2
3     ruleflow-group "REA_E-"
4
5     when
6         $q : Question(answer.value=="true",
7             genericQuestion.name==GQuestions.AGREEMENT_IN_THE_PROCESS);
8
9     then
10        System.out.println("\r Starting: " + drools.getRule().getName());
11
12        #Remove the REA Agent identification business events from the business process
13        EMFREAViewHandler.removeEventsByTypeName( process,
14            "REAAgentIdentification",
15            $q.param(0));
16
17        System.out.println(drools.getRule().getName() + " executed successfully");
18    end

```

Figure 6.32 Exemple de règle de transformation de la vue REA.

La Figure 6.33 illustre le code de la méthode `removeEventsByTypeName` (ligne 13) de la classe utilitaire `EMFREAViewHandler`. Cette méthode est utilisée dans la règle de transformation de la vue REA de la Figure 6.32 pour supprimer les événements liés à l'identification de l'agent partenaire. La méthode `EcoreUtil.getObjectsByType` (`Collection objects, EClassifier type`) (ligne 18) identifie toutes les instances du type passé en paramètre. Dans notre cas, elle retourne tous les événements REA qui servent à identifier le partenaire d'affaires (i.e. le type est `REAAgentIdentification`).

L'instruction `EcoreUtil.delete(o, true)` (ligne 27) supprime l'événement REA (i.e. l'objet *o*) de la ressource EMF (i.e. la vue REA) qui le contient. Elle supprime aussi toutes les références à cet événement depuis l'ensemble des ressources EMF (*ResourceSet*) (i.e. les autres vues du processus). Le paramètre `true` s'assure de supprimer, d'une manière récursive, tous les éléments exclusifs à l'événement. Par conséquent, les ressources REA liées à chacun de ces événements seront aussi retirées de la vue REA.

```

1  /**
2   * Removes process REA events based on the type name.
3   * @param bProcess
4   * @param typeName
5   * @param processName
6   * @throws ViewElementNotFoundException
7   */
8   public static void removeEventsByTypeName(BProcess bProcess,
9                                           String typeName,
10                                          String processName)
11   throws ViewElementNotFoundException {
12
13       // get the process
14       Process process = findProcessByName(bProcess.getREAView().getView(),
15                                          processName);
16
17       System.out.println("Get all process REA events by type: " + typeName);
18
19       Collection<REAEvent> objects = EcoreUtil.getObjectsByType(
20                                           process.getEvent(),
21                                           BpPackage.eINSTANCE.getEClassifier(typeName));
22
23       if (objects.isEmpty())
24           throw new ViewElementNotFoundException(
25               "Could not find any REA event with the type : " + typeName);
26
27       for (REAEvent o : objects) {
28           System.out.println("REA Event to remove: " + o.getName());
29           EcoreUtil.delete(o, true);
30       }
31   }

```

Figure 6.33 Exemple EMF : la méthode removeEventsByTypeName.

6.3.2 Le flux des transformations

Nous avons montré dans la section 6.2.1 qu'il n'y a pas de chevauchement entre les vues d'un processus d'affaires. Nous avons aussi montré que l'ordre de transformation des vues pour la même question dépend de la transformation de la vue REA. En effet, la vue dynamique, la vue organisationnelle et la vue informationnelle dépendent de la vue REA.

Nous avons alors défini un ordre d'exécution des règles de transformation des vues pour la même question (i.e. suite à une réponse à une seule question). Nous distinguons deux types de séquences correspondant à l'ajout (i.e. T^{rea+}), et à la suppression des événements REA (i.e. T^{rea-}). En effet, lors de l'ajout d'un élément fonctionnel au processus (i.e. événement ou ressource REA), on l'ajoute d'abord dans la vue REA, puis on ajoute les références dans les

autres vues. Pour le retrait, on retire les références aux éléments de la vue REA d'abord, puis on supprime l'élément.

Nous avons choisi de mettre en œuvre ces séquences de transformation avec le système de gestion de flux *Drools Flow* (actuellement *jBPM*). *Drools Flow* offre les avantages suivants (JBoss, 2012):

- une intégration avancée de processus¹ et de règles. *Drools Flow* permet aux organisations de définir certaines règles d'affaires de leurs processus d'affaires. De plus, il offre une interface de programmation unifiée dite *Knowledge API* et des outils qui permettent aux utilisateurs de spécifier, exécuter et contrôler les règles et les processus;
- une modélisation déclarative; et
- un moteur générique de processus qui supporte plusieurs langages de processus comme WS-BPEL, OSWorkflow et jPDL qui est le langage de processus défini par jBPM.

La Figure 6.34 et 6.35 montre le diagramme *Drools Flow* des séquences $T^{\text{rea}+}$ et $T^{\text{rea}-}$ respectivement. Une étape du diagramme *Drools Flow* (i.e processus) correspond à un groupe de flux (non visible dans le diagramme). Chaque règle Drools fait éventuellement partie d'un groupe de flux. Par exemple, la règle de la Figure 6.32 appartient au groupe *REA_E-* qui correspond à la première étape de la séquence $T^{\text{rea}-}$.

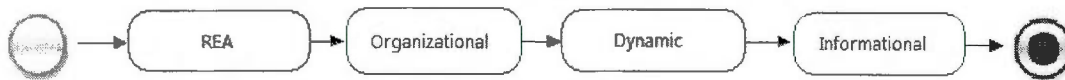


Figure 6.34 Le flux de la séquence $T^{\text{rea}+}$ avec Drools Flow.



Figure 6.35 Le flux de la séquence $T^{\text{rea}-}$ avec Drools Flow.

¹Une séquence de règles dans *Drools Flow* correspond au concept de processus.

La Figure 6.36 montre des extraits du code qui permet de charger les règles de transformation ainsi que le flux de transformation $T^{\text{rea+}}$. D'abord, il faut créer une instance de la classe `KnowledgeBuilder`. Par la suite, il faut ajouter les règles et le flux à cette instance. L'ajout de règles est identique à l'ajout du flux, seul le type de la ressource change de *drl* à *drf*. Enfin, il faut créer une instance de la base de connaissances à partir de la classe de fabrication `KnowledgeBaseFactory` et y ajouter les règles de transformation et le flux de transformation $T^{\text{rea+}}$.

```
private static KnowledgeBase readKnowledgeBase() throws Exception {
    // 1. Creating a Knowledge Base
    KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();

    // Add the Rules for the question 1 to the Knowledge Base.
    kbuilder.add(ResourceFactory.newClassPathResource("q1/DynamicView.drl"),
        ResourceType.DRL);
    kbuilder.add(ResourceFactory.newClassPathResource("q1/REAView.drl"),
        ResourceType.DRL);
    kbuilder.add(ResourceFactory.newClassPathResource("q1/OrganizationalView.drl"),
        ResourceType.DRL);
    kbuilder.add(ResourceFactory.newClassPathResource("q1/InformationalView.drl"),
        ResourceType.DRL);

    // add the process to the Knowledge Base. Note that this is almost identical
    // to adding rules to the Knowledge Base
    kbuilder.add(ResourceFactory.newClassPathResource("/TransformationsFlowREA+.rf"),
        ResourceType.DRF);
    ...
    ...
    KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
    kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
    return kbase;
}
```

Figure 6.36 Extrait du code qui permet de charger les règles et le flux $T^{\text{rea+}}$.

```

// load up the knowledge base
KnowledgeBase kbase = readKnowledgeBase();
// create a stateful Knowledge Session
StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
....
....
// start the rea+ transformation process
ksession.startProcess("ca.uqam.latece.bpsf.transformation.process.rea+");
// fire the rules
ksession.fireAllRules();
....
....
// dispose the rule session
ksession.dispose();

```

Figure 6.37 Extrait du code permettant le déclenchement du processus de spécialisation.

Pour exécuter les règles de transformation selon le flux $T^{\text{rea+}}$, il suffit de créer une session de type `StatefulKnowledgeSession`. Pour lancer un processus de règles (i.e. un diagramme *Drools Flow*), il suffit d'appeler la méthode `startProcess` de cette session. Le paramètre de cette méthode correspond à l'identificateur du processus de règles à lancer. Dans notre cas, `ca.uqam.latece.bpsf.transformation.process.rea+` qui correspond à la séquence $T^{\text{rea+}}$. Pour lancer les règles, il faut appeler la méthode `fireAllRules` à partir de l'instance de `StatefulKnowledgeSession`. Il faut noter qu'on peut lancer plusieurs processus de règles. La Figure 6.37 montre des extraits du code qui permettent de lancer les règles et les flux de transformation.

6.4 Traitement de transformations en chaîne

Dans cette section, nous abordons deux aspects importants des approches transformationnelles. Nous discutons de la problématique de génération de modèles intermédiaires et du chevauchement de règles de transformations dans le cas d'une transformation globale (i.e. associée à plusieurs questions/réponses) (sect. 6.2.1.3).

6.4.1 Doit-on générer des modèles intermédiaires ?

Notre approche de spécialisation utilise des règles de transformation qui s'appliquent aux vues du processus d'affaires. Chaque vue représente un modèle. Nous avons le choix de générer des modèles intermédiaires après chaque transformation unitaire (i.e. après la réponse à chaque question), ou seulement le modèle final. Nous distinguons alors deux situations. D'abord, les modèles intermédiaires sont conservés dans le catalogue de processus d'affaires. Deuxièmement, ces modèles sont ajoutés à la base de faits et provoquent des changements au niveau des règles (e.g. le déclenchement d'une nouvelle règle).

Générer des modèles intermédiaires revient à ajouter au catalogue un processus d'affaires après chaque question/réponse. Ceci présente deux inconvénients. D'abord, cela encombre le catalogue, ce qui pose principalement un problème d'indexation et de recherche. Deuxièmement, cela risque de générer de nombreux processus dont certains sont sans importance pour les organisations. Toutefois, le fait de générer des modèles intermédiaires présente un avantage durant la phase de découverte du processus le plus proche. En effet, l'analyste a plus de chances de retrouver un processus qui répond aux besoins de son organisation ou un processus très proche, du fait que le nombre de processus d'affaires dans le catalogue est important. Parallèlement, quand on génère les modèles finaux seulement, le catalogue est plus léger. Cependant, l'analyste a moins de chances de retrouver un processus qui répond à ses besoins, ou celui qui est le plus proche. En effet, pour ce dernier point, il est possible que le chemin de spécialisation recherché soit une sous-séquence d'un autre chemin à partir du même processus et que malheureusement, on ne puisse pas l'utiliser. Par exemple, supposons que le catalogue contienne un processus P1 et un processus P2 spécialisé à partir de P1 par le chemin de spécialisation $\langle (q_1, r_1), (q_2, r_2), \dots, (q_i, r_i) \dots (q_n, r_n) \rangle$. Si l'analyste cherche un processus avec le chemin de spécialisation $\langle (q_1, r_1), (q_2, r_2), \dots, (q_i, r_i) \rangle$ à partir de P1, il ne pourra bénéficier de la spécialisation de P2. Il devra alors chercher un autre chemin ou refaire le processus de spécialisation à partir de P1. Nous pensons que le choix de l'une ou l'autre stratégie est avant tout un problème métier et devrait donc être traité selon cette perspective.

Le deuxième point est relié au mode d'exploitation non monotone dans les approches transformationnelles. En effet, lorsque les modèles intermédiaires sont considérés dans le processus de transformation, cela peut provoquer, par exemple, le déclenchement de nouvelles

règles. Prenons notre exemple de la section 6.2.2. Après la sous-traitance du processus de distribution, un nouveau processus d'échange est ajouté à la chaîne de valeur. Nous devons donc reposer la question du calcul automatique de la créance pour ce dernier processus.

6.4.2 Les transformations chevauchantes : Problème ouvert

Le chevauchement des règles de transformation est un problème commun aux approches de transformation de modèles. Le chevauchement se produit quand les règles de transformation manipulent les mêmes instances d'un modèle. Dans ce cas, les transformations peuvent donner lieu à un problème de non-confluence¹.

Nous avons rencontré ce problème lors de l'application de règles de transformation associées à plusieurs questions/réponses (i.e. le cas d'une transformation globale) (sect. 6.2.1.3). Le problème de chevauchement des règles de transformation est souvent relié au concept d'indépendance parallèle des règles. Deux règles de transformation sont dites parallèlement indépendantes si toutes les applications possibles de ces deux règles ne s'entravent pas les unes les autres. Si deux règles ne sont pas parallèlement indépendantes, elles peuvent donner lieu à un problème de chevauchement. Dans ce cas, l'ordre d'exécution de ces règles peut poser problème. Par conséquent, le processus obtenu après l'exécution de ces règles dépend de la séquence dans laquelle elles sont appliquées. Nous discutons de la confluence de transformations avec plus de détails dans le chapitre 7.

6.5 Conclusion

Nous avons présenté, dans ce chapitre, une nouvelle approche de spécialisation de processus d'affaires. Nous avons montré comment adapter un processus générique aux besoins spécifiques des organisations.

Notre démarche de spécification de processus d'affaires est basée sur l'ontologie d'affaires REA. Elle utilise une approche par questions. Les questions et les transformations de

¹La propriété de confluence exige que l'application des règles de transformation sur un modèle donne toujours le même résultat ou un résultat équivalent quelque soit l'ordre d'application de ces règles (Küster et Abd-El-Razik, 2006).

spécialisation que nous proposons sont génériques puisqu'elles peuvent s'appliquer à des processus d'affaires provenant de différents domaines. Nous croyons que notre approche peut être assimilée par des analystes d'affaires sans grande expertise informatique/technique car nos questions ont une sémantique métier. En effet, toutes nos questions sont basées sur des patrons d'affaires de l'ontologie REA.

Nous avons montré comment représenter et mettre en œuvre les questions et les règles de transformation. Nous avons aussi expliqué que la transformation d'un processus d'affaires passe par la transformation de ses vues, soit les vues REA, organisationnelle, dynamique et informationnelle.

Nous avons aussi soulevé la problématique de l'ordre d'exécution des règles de transformation. Nous avons montré qu'il est possible de déterminer un ordre d'exécution des règles de transformations entre les vues pour une réponse à une seule question. Toutefois, comme nous l'avons souligné à la section 6.4, notre approche de transformation fait face au problème des transformations qui s'appliquent entre plusieurs questions. Dans ce cas, nous parlons de règles de transformation chevauchantes. Nous discutons de cet aspect dans le chapitre suivant qui sera dédié à la validation de notre approche.

CHAPITRE VII

EXPÉRIMENTATION ET VALIDATION

Dans les chapitres précédents, nous avons présenté notre approche de réutilisation et d'adaptation de processus d'affaires. Nous avons proposé une approche transformationnelle qui permet le passage d'un modèle générique de processus d'affaires à un modèle spécifique à une organisation. Nous avons présenté, dans les chapitres 3, 5 et 6, plusieurs exemples qui démontrent le fonctionnement de notre approche. Ce chapitre sera dédié à l'expérimentation et la validation. Pour cela, il fallait d'abord définir les différents aspects que nous devons valider. Ensuite, pour chaque aspect, nous devons identifier les approches de validation possibles. Par la suite, nous avons réalisé un ensemble de tests en utilisant plusieurs processus d'affaires provenant de plusieurs domaines. Il fallait aussi considérer l'application de plusieurs questions génériques et valider plusieurs règles de transformation. Pour valider certains aspects, nous avons eu recours à trois experts dans le domaine des processus d'affaires. Le premier expert est un spécialiste dans les systèmes PGI. Le deuxième est un consultant principal en impartition de processus d'affaires et spécialiste de l'approche BPM. Le troisième expert est un professeur universitaire avec une très grande expérience industrielle en gestion de processus d'affaires.

Notre analyse a identifié les aspects de validation suivants : (i) la représentation de processus d'affaires, (ii) l'applicabilité des questions, et (iii) les transformations de spécialisation. À la section 7.1, nous discutons plus en détails du besoin de valider ces aspects. La section 7.2 présente les données expérimentales. La section 7.3 traite de la validation de la représentation de processus d'affaires. Nous présentons la validation de

l'aspect relié à l'application des questions génériques à la section 7.4. Par la suite, nous présentons la démarche que nous avons adoptée pour valider l'aspect transformation à la section 7.5. Nous terminons par une conclusion à la section 7.6.

7.1 Aspects à valider

Pour valider notre approche de spécification de processus, nous avons analysé toutes les étapes de notre démarche. Pour chacune des étapes, nous nous sommes posés la question sous-jacente à sa validation. Notre analyse a permis d'identifier les aspects suivants :

1. *La validation de la représentation de processus d'affaires* : Cela nous amène à vérifier la capacité de notre méta-modèle à représenter les processus d'affaires. Le but serait de vérifier : (i) que l'on peut représenter un ensemble utile et représentatif de processus d'affaires provenant de différents domaines, et (ii) que notre représentation produit des modèles de processus corrects au niveau syntaxique (incluant les contraintes d'affaires) et sémantique.
2. *La validation de l'applicabilité des questions génériques* : Cet aspect consiste à vérifier : (i) que les questions génériques identifiées par notre méthodologie peuvent être appliquées à des processus d'affaires issus de différents domaines, et (ii) que l'application des questions spécifiques (i.e. les instances des questions génériques) dans le contexte de ces processus d'affaires préserve la sémantique des questions génériques.
3. *La validation des transformations élucidées* : Cet aspect clé consiste à valider les transformations qui sont la base de notre approche de spécialisation de processus d'affaires. Dans le contexte transformationnel de notre approche, nous nous sommes posés plusieurs questions reliées à la correction de nos transformations. Nous avons alors décidé de vérifier si nos transformations génèrent des modèles syntaxiquement et sémantiquement corrects. Toutefois, nous nous sommes aussi penchés sur les questions de confluence des transformations.

Tous ces aspects doivent être vérifiés dans le contexte de plusieurs processus d'affaires provenant de différents domaines.

7.2 Données expérimentales

Pour valider notre approche, nous avons considéré différents processus d'affaires d'échange et de conversion provenant de différents domaines d'affaires. Nous avons étudié plusieurs catalogues de processus, notamment le manuel MIT (Malone *et al.*, 1999), la chaîne de valeur de Michael Porter (1975), le manuel de Pavel Hruby (2006) et le catalogue du progiciel SAP (Curran, Keller et Ladd, 1998; Murray, 2009; Magal et Word, 2010). Ces catalogues présentent principalement les mêmes processus d'affaires. À la fin, nous avons considéré huit processus d'affaires que nous avons classifiés selon les processus clés des systèmes PGI tels que montrés par Magal et Word (2010) dans la Figure 7.1. Nous avons étudié les processus de vente et distribution, d'achat, de production, de gestion d'actifs et service à la clientèle, de gestion du capital humain et finalement, de marketing et finance. Le tableau 7.1 énumère les huit processus d'affaires que nous avons utilisés pour valider notre approche. Nous montrons aussi la classification selon la chaîne de valeur de Porter (1975).

Pour vérifier les aspects d'applicabilité des questions génériques et des transformations (aspects 2 et 3), nous avons étudié un ensemble de sept questions relatives aux patrons d'affaires de Hruby (2006). Ces questions sont présentées dans la section 7.4.

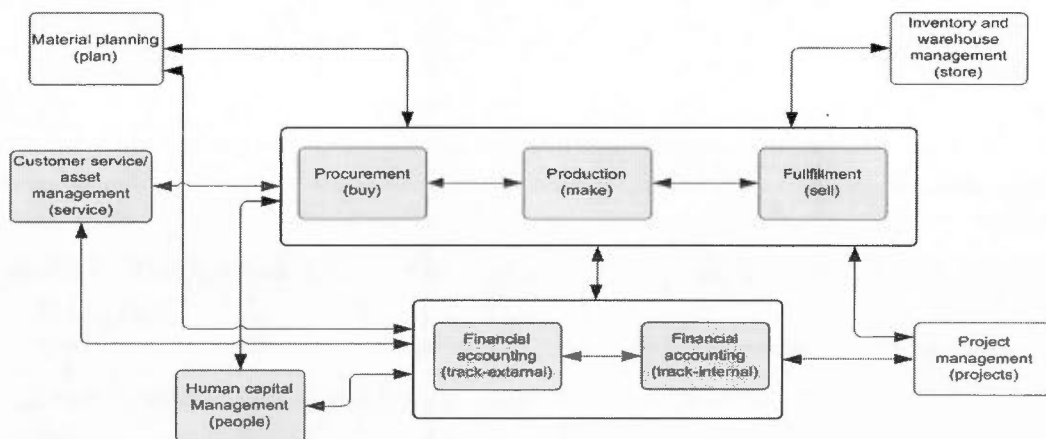


Figure 7.1 Les processus d'affaires clés des systèmes PGI. (Source : Magal et Word, 2010)

Tableau 7.1

Les processus d'affaires utilisés pour valider notre approche

Processus d'affaires	Classification selon les processus PGI	Classification selon la chaîne de valeur de Porter
Approvisionnement	Achat	Logistique entrante
Production et inspection	Production	Les opérations
Vente et distribution	Vente	Logistique sortante
Maintenance et service à la clientèle	Service	Service
Recrutement	Gestion du capital humain	Ressources humaines
Prêt Financier	Finance	Marketing et vente
Vente de police d'assurance	Vente	Marketing et vente
Gestion de la paie	Gestion du capital humain	Ressources humaines

7.3 Validation de la représentation de processus d'affaires

Cette validation, réalisée en deux étapes, consiste à vérifier la capacité de notre méta-modèle à représenter les processus d'affaires. D'abord, nous vérifions que l'on peut représenter un ensemble utile de processus d'affaires provenant de différents domaines. Deuxièmement, nous vérifions que notre représentation est correcte au niveau syntaxique et sémantique. Pour cet aspect, nous avons considéré les processus du Tableau 7.1 ainsi que d'autres processus du manuel de Pavel Hruby (2006). En tout, nous avons représenté dix processus d'affaires.

7.3.1 Vérification de la capacité de représentation

La représentation de processus d'affaires consiste à modéliser un processus générique selon le processus de représentation présenté dans le chapitre 5 (sect. 5.1). Ce processus générique en entrée peut être représenté sous différents formats. Pour montrer notre approche, nous avons choisi la norme BPMN pour sa large utilisation et son acceptation dans la communauté. En parallèle, notre approche de spécification de processus d'affaires n'impose pas de stratégie particulière pour le choix de représentation de processus génériques (i.e. la stratégie du processus maximal, du processus minimal, du processus standard, du processus le plus utilisé et du processus ayant la distance moyenne minimale) (sect. 5.1).

L'aspect de validation de la représentation dépend de la capacité de notre méta-modèle à modéliser les quatre vues de processus d'affaires, soit : *la vue REA étendue*, *la vue dynamique*, *la vue informationnelle* et *la vue organisationnelle*. Nous avons présenté dans le chapitre 5 la mise en œuvre de notre méta-modèle. Ce dernier a été construit en étapes. D'abord, nous avons commencé par implémenter le méta-modèle de l'ontologie REA que nous avons augmenté pour supporter les processus d'affaires de nos jours. Cette augmentation consistait en l'ajout des événements d'affaires et informationnels, ainsi que certaines extensions organisationnelles telles que les concepts qui permettent la distinction entre l'agent interne et externe (extensions BPDM) et l'ajout de l'agent tiers (*Third party*). Le modèle REA obtenu représente le méta-modèle de la vue REA et de la vue organisationnelle. Par la suite, nous avons complété notre méta-modèle par une partie du modèle comportemental (*Behavior model*) de BPDM. Cette partie représente le méta-modèle de la vue dynamique. Finalement, la vue informationnelle, qui représente la structure des ressources, a été modélisée avec EMF ecore sous forme d'un diagramme de classe UML. C'est ainsi qu'on arrive à modéliser les quatre vues d'un processus d'affaires.

Notre processus de représentation de processus d'affaires consiste en une étape d'analyse suivie d'une étape de conception. L'étape d'analyse sert à décomposer le processus d'affaires en entrée en chaîne de valeur REA. Par la suite, il faut analyser tous les événements de chaque processus élémentaire et le classer en événement économique, d'affaires ou informationnel. L'étape de conception consiste à modéliser chaque processus d'affaires identifié dans l'étape d'analyse selon le patron REA d'échange ou de conversion.

Par la suite, il faut concevoir la vue REA, la vue dynamique, la vue informationnelle et la vue organisationnelle.

Nous avons réalisé l'exercice de modélisation avec l'ensemble des processus énumérés dans le Tableau 7.1. Nous avons même fait l'exercice avec d'autres processus (voir le Tableau 7.2) du manuel de Pavel Hruby (2006). Nous étions capables de modéliser chacun de ces processus sans grandes difficultés. Le résumé des résultats est présenté dans le Tableau 7.2 où « Oui » indique que nous étions capables de modéliser la vue en question.

Cette expérience nous a permis de constater que notre méta-modèle est assez complet pour permettre de modéliser un ensemble utile de processus d'affaires. Cependant, certaines difficultés que nous avons rencontrées, lors de la représentation de processus génériques, nous ont permis de le compléter. Voici les trois difficultés que nous avons affrontées :

1. *La classification des étapes du processus d'affaires* : Chaque étape de chacun des processus de la chaîne de valeur doit être classifiée en événement REA. Cette classification est rendue possible grâce à l'ajout des événements d'affaires. En effet, chaque étape, qui n'est pas un événement économique ou informationnel, peut être classifiée en un événement d'affaires. Cependant, il fallait analyser et écarter les événements informationnels qui n'ont pas d'incidence sur le processus de spécialisation.
2. *La modélisation de la vue dynamique* : Cette difficulté est liée à la complétude de notre méta-modèle relativement aux événements de contrôle tels que les différents événements intermédiaires et les passerelles (*Gateway* en BPDM). Pour ce dernier point, nous avons ajouté quelques extensions BPMN telles que définies dans la spécification de BPDM (OMG, 2008).
3. *Le passage de la notation BPMN à la vue REA* : Notre expérience de modélisation de la vue REA nous a montré que l'étape de classification des activités du processus BPMN en événements REA est transparente et sans pertes. La décision de ne conserver que les événements informationnels ayant une incidence sur le processus d'affaires est la nôtre (voir la section 5.1 du chapitre 5). Cependant, nous avons

remarqué que les événements REA duaux pour les processus de conversion ne sont pas toujours présents dans le processus BPMN. Cela est tout à fait normal car, en général, les concepteurs de processus d'affaires font abstraction de certains détails, notamment ceux reliés à la consommation et l'utilisation de certaines ressources économiques. Par exemple, dans un processus de production, l'analyste d'affaires fera abstraction des outils utilisés ou de la main-d'œuvre consommée même si ces ressources sont économiques. Comme les modèles REA exigent que chaque événement d'incrément doit être relié, via la relation de *dualité*, à au moins un événement de décrément et inversement, nous avons deux choix : (i) de montrer l'événement économique dual dans le processus générique (i.e. le processus BPMN) ou (ii) d'ajouter cet événement directement dans le vue REA.

7.3.2 Validation de la correction des modèles

Le deuxième point de validation de la représentation de processus d'affaires consiste à vérifier la correction syntaxique et sémantique du modèle obtenu après l'application du processus de modélisation.

Pour vérifier la correction syntaxique, nous nous sommes posés la question : *''Est ce que le processus obtenu est conforme à notre méta-modèle ? ''*. Nous avons validé ce point en deux étapes. La première s'assure que la technique de construction du modèle (les vues) ne produit qu'une instance de notre méta-modèle. La deuxième consiste à vérifier que le modèle obtenu est une instance valide de notre méta-modèle. Cette stratégie de validation a fonctionné et a donné satisfaction pour l'ensemble des processus de notre référentiel. En effet, pour la première étape, nous avons conçu les vues REA, dynamique et organisationnelle avec un éditeur fourni par le cadre EMF. Cet éditeur permet de garantir, par construction, que l'instance obtenue est une instance de notre méta-modèle. En parallèle, la vue informationnelle utilise un éditeur de diagramme *ecore* pour UML. Pour la deuxième étape, nous avons utilisé le cadre de validation d'EMF (*EMF Validation Framework*) qui, à partir d'un modèle, retourne un diagnostique positif ou négatif dépendamment de la validité

du modèle relativement à son méta-modèle. Ce cadre de validation permet aussi de valider les règles d'affaires associées au méta-modèle (voir appendice C).

Tableau 7.2

Résultat de la validation de l'aspect de représentation de processus d'affaires

Processus d'affaires	Vue Informationnelle	Vue Fonctionnelle	Vue Dynamique	Vue Organisationnelle
Logistique d'approvisionnement	Oui	Oui	Oui	Oui
Production et inspection	Oui	Oui	Oui	Oui
Vente et distribution	Oui	Oui	Oui	Oui
Maintenance et service à la clientèle	Oui	Oui	Oui	Oui
Recrutement	Oui	Oui	Oui	Oui
Prêt financier	Oui	Oui	Oui	Oui
Vente de police d'assurance	Oui	Oui	Oui	Oui
Gestion de la paie	Oui	Oui	Oui	Oui
Location de ressources	Oui	Oui	Oui	Oui
Marketing et publicité	Oui	Oui	Oui	Oui

La correction sémantique du modèle de processus obtenu après la modélisation consiste à vérifier si le modèle obtenu préserve la sémantique du modèle BPMN en entrée. En d'autres termes, notre objectif est d'établir l'équivalence sémantique entre les deux modèles (i.e. le modèle de processus obtenu après la modélisation et le modèle BPMN en entrée). Contrairement à la vérification syntaxique, nous ne pouvons prouver de manière systématique l'équivalence sémantique entre le modèle source en BPMN et le modèle obtenu après l'application de notre processus de modélisation. Nous avons donc demandé à nos experts dans le domaine d'affaires d'analyser manuellement les modèles obtenus pour vérifier si la sémantique du modèle en entrée a été préservée. Après 4 heures de travail, notre premier expert a pu analyser les modèles XMI de huit processus d'affaires. La réponse a donné satisfaction pour l'ensemble des processus d'affaires analysés. Notons que la structure des ressources n'a pu être validée puisqu'elle n'est pas visible dans le processus d'affaires source (i.e. le processus BPMN). Notons aussi que l'expert a reconnu que la décomposition de processus d'affaires en vues (i.e. les vues REA, dynamique, organisationnelle et informationnelle) permet de bien l'interpréter.

Le résumé des résultats de la validation de cet aspect est présenté dans le tableau 7.3. « Oui » indique que la vérification en question a été complétée avec succès.

À partir de ces résultats, nous estimons que notre méta-modèle, illustré dans l'appendice C, nous a permis de valider notre approche de représentation de processus d'affaires positivement.

Tableau 7.3

Résultat de la validation de la correction des modèles obtenus

Processus d'affaires	Vérification syntaxique	Vérification sémantique
Logistique d'approvisionnement	Oui	Oui
Production et inspection	Oui	Oui
Vente et distribution	Oui	Oui
Maintenance et service à la clientèle	Oui	Oui
Recrutement	Oui	Non validée
Prêt financier	Oui	Oui
Vente de police d'assurance	Oui	Oui
Gestion de la paie	Oui	Oui
Location de ressources	Oui	Non validée
Marketing et publicité	Oui	Oui

7.4 Validation de l'applicabilité des questions génériques

Cet aspect consiste à vérifier deux points : (i) que les questions génériques peuvent être appliquées à différents processus d'affaires, et (ii) que l'application de ces questions a du

sens (i.e. la sémantique de la question est préservée) dans le contexte de ces processus. Ce dernier point a été validé par des experts dans le domaine.

7.4.1 Identification des questions génériques

Pour vérifier l'applicabilité des questions génériques, nous avons considéré les questions formulées à partir des patrons suivants : (i) Le patron d'échange, (ii) le patron de sous-traitance, (iii) le patron de contrat, (iv) le patron d'engagement, (v) le patron de matérialisation de la créance, et (vi) le patron d'historiques de transactions.

- *Les questions du patron d'échange* : Selon le patron d'échange, les événements économiques duaux peuvent se produire dans un ordre indéfini. Ils peuvent aussi se produire au même moment, auquel cas, nous parlons d'échange instantané. Le concept de permutation d'événements duaux est illustré dans la Figure 7.2. Dans le premier cas, l'événement de décrétement précède l'événement d'incrément. Dans le deuxième cas, l'ordre est inversé. Finalement, dans le troisième cas, les deux événements se produisent au même moment.

De là, nous envisageons les questions suivantes :

1. *Les échanges du processus d'affaires $\{0\}$ sont ils instantanés ?* (Q1)

Dans le cas d'une réponse positive à cette question, aucun déséquilibre n'est créé au niveau de la valeur des ressources économiques échangées. Nous n'avons alors pas besoin de créance. Rappelons que la créance économique existe seulement lorsqu'un laps de temps sépare les événements économiques duaux (i.e. l'échange n'est pas instantané). Dès lors, la matérialisation, l'envoi et la vérification de la créance ne sont pas nécessaires.

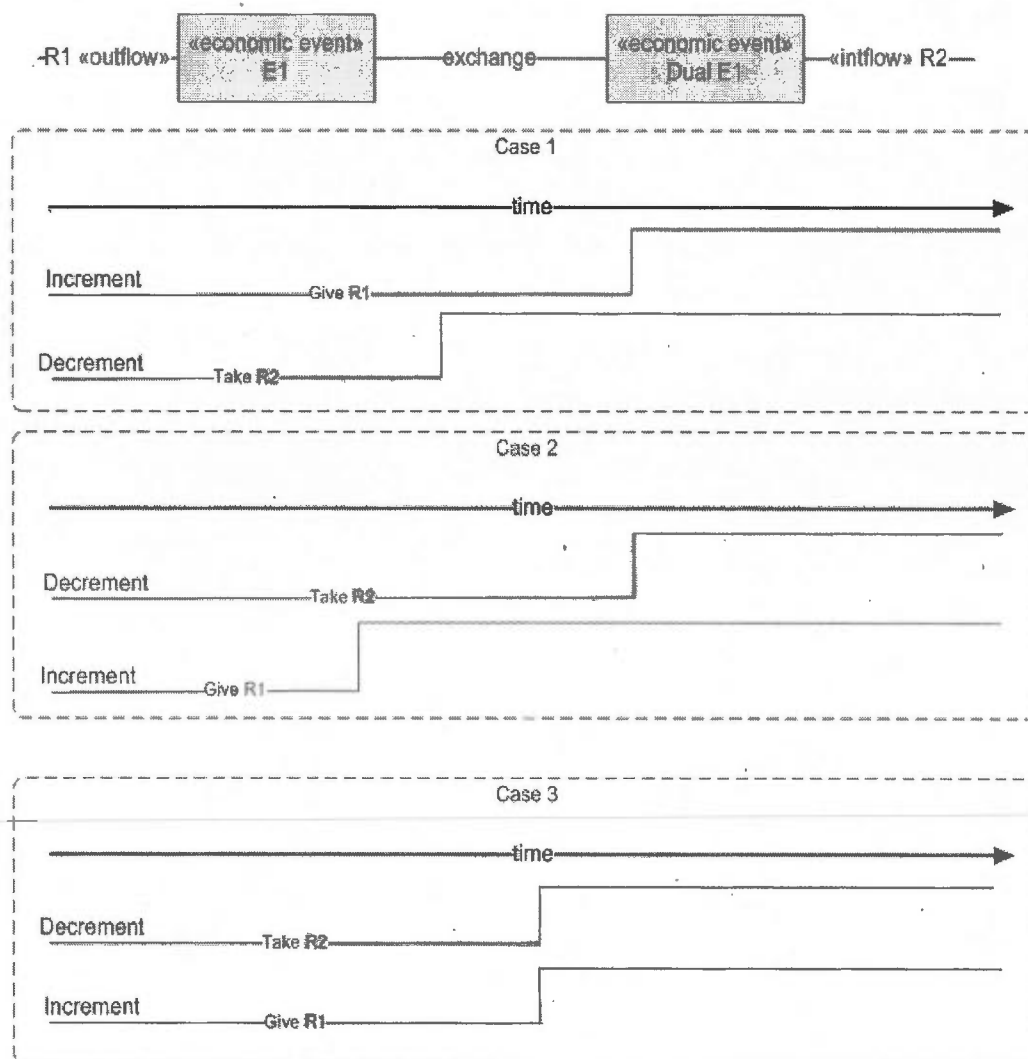


Figure 7.2 Les événements économiques dans un processus d'échange. (Source : Hruby, 2006)

2. Est-ce que [les événements économiques d'incrément] {0} précède(nt)[les événements économiques de décrétement] {1} pour le processus d'échange {2} ? (Q2)

Le texte entre crochets permet de donner un sens à la question générique. Ce texte sera supprimé lors de l'instanciation. Par conséquent, il n'apparaîtra pas dans la question spécifique. Cette question comprend un prédicat qui permet de définir la

relation entre les paramètres. Par exemple, le prédicat $P(X, Y) = X \text{ in } Y$ qui est vrai si X est un sous-ensemble de Y . Pour cette question, la proposition suivante doit être vraie :

$P(\text{param}(0), \text{param}(2).\text{incrementEvents}())$ ET

$P(\text{param}(1), \text{param}(2).\text{decrementEvents}())$ où $\text{param}(i)$

représente le $i^{\text{ème}}$ paramètre de la question.

Dans le cas d'une réponse positive et que les événements d'incrément ne précédaient pas les événements de décrétement, il faut inverser la séquence des événements économiques.

- *La question du patron de sous-traitance* : Nous avons présenté le patron de sous-traitance de processus de la chaîne de valeur dans la section 6.1.2 du chapitre 6. Nous avons proposé la question: *Est-ce que l'organisation prévoit la sous-traitance du processus {0}* (Q3)
- *La question du patron de contrat* : Cette question traite de l'existence d'accord qui réglemente le déroulement du processus d'affaires (sect. 6.1.2). Nous avons proposé la question: *Est-ce que le déroulement du processus {0} est régi par un accord ?* (Q4)
- *La question du patron d'engagement* : À partir du patron d'engagement, nous pouvons examiner si l'échange est régi par des obligations futures (sect. 6.1.2). Par conséquent, nous proposons la question : *Est-ce que le déroulement du processus d'affaires {0} est régi par des engagements ?* (Q5)
- *La question du patron de matérialisation de la créance* : Cette question traite du calcul automatique de la valeur de la créance. Nous avons alors suggéré, dans la section 6.1.2 (chap. 6), la question : *Est-ce que le processus {0} supporte le concept de Two-way match pour la gestion des créances ?* (Q6)
- *La question du patron d'historiques* : Cette question est reliée à l'historique des transactions concernant les événements économiques, les contrats et les engagements avec les partenaires (*Posting pattern*) (Hruby, 2006). Nous proposons la

question: *Est-ce que les informations concernant les transactions entre les partenaires du processus d'affaires {0} sont mémorisées ?* (Q7)

Comme nous l'avons expliqué dans le chapitre 6 (sect. 6.2), nous ne considérons pas les patrons d'affaires qui génèrent des questions spécialisant seulement la vue informationnelle. Ce type de transformations est relativement bien connu et traité dans les approches orientées objet. À partir de là, nous avons exclu les questions reliées à certains patrons de Hruby (2006) tels que le patron de réconciliation des créances (*Reconciliation pattern*), le patron de classification des entités REA (*Classification pattern*), le patron d'identification des entités (*Identification pattern*) et le patron d'informations agrégées (*Account pattern*).

7.4.2 Généricité/Pertinence des questions génériques

Nous avons appliqué l'ensemble des questions génériques identifiées dans la section 7.4.1 aux processus d'affaires énumérés dans le tableau 7.1. Nous avons consulté deux spécialistes dans le domaine de processus d'affaires pour vérifier la validité des questions génériques. Nous avons alors instancié les questions génériques pour chacun des processus d'affaires. Par la suite, nous avons demandé aux spécialistes si les questions instanciées (i.e. spécifiques) s'appliquent et préservent leur sémantique d'affaires dans le contexte de chacun de ces processus d'affaires. Les résultats obtenus sont présentés dans le tableau 7.4. « Oui » indique que la question est valide (i.e. s'applique et préserve sa sémantique métier) pour le processus générique de notre référentiel. Notons que cette même question ne s'applique pas nécessairement à un autre processus d'affaires de la même famille après certaines spécialisations (i.e. après avoir répondu à certaines questions). Ce point est discuté après la présentation des résultats du tableau 7.4. « Non » indique que la question spécifique n'est pas pertinente dans le contexte du processus. « NA » indique que la question générique ne s'applique pas à ce type de processus d'affaires (e.g. la question s'applique seulement à un processus d'échange ou la question s'applique seulement à un sous-processus). Nous utilisons « En partie » dans le cas où la question est valide seulement pour certains processus obtenus après la décomposition du processus d'affaires initial en une chaîne de valeur.

Tableau 7.4

Résultat de la validation de l'applicabilité des questions génériques

Processus d'affaires	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Logistique d'approvisionnement	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Production et inspection	NA	NA	Oui	NA	Oui	NA	Oui
Vente et distribution	En partie	En partie	Oui	En partie	Oui	En partie	Oui
Maintenance et service à la clientèle	En partie	En partie	Oui	En partie	Oui	En partie	Oui
Recrutement	Non	Oui	Oui	Oui	Oui	Oui	Oui
Prêt financier	Non	Oui	Oui	Oui	Oui	Oui	Oui
Vente de police d'assurance	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Gestion de la paie	Non	Oui	Oui	Oui	Oui	Oui	Oui

Durant cet exercice, les experts ont soulevé le fait que certaines questions ne devraient plus être posées après avoir appliqué d'autres questions (i.e. après des spécialisations). En effet, prenons l'exemple des questions Q1 et Q6. Après l'application de la règle de transformation correspondante à une réponse positive à la question Q1, l'échange devient instantané. Dès lors, nous n'avons pas besoin de créance. Ainsi, la matérialisation, l'envoi et la vérification de la créance ne sont pas nécessaires. Par conséquent, la question Q6 qui traite de la matérialisation de la créance n'est plus valide dans ce contexte. Le même raisonnement s'applique à la question Q4 et Q5 dans le cas d'une réponse positive à la question Q4 (i.e. question du patron contrat) et une réponse négative à la question Q5 (i.e.

question du patron d'engagement). En effet, le contrat REA doit régler les obligations futures (i.e. les engagements) (Hruby, 2006).

Pour les deux exemples, nous remarquons que l'ordre d'application des questions n'est pas important. Dès lors, la deuxième question n'est pas applicable après avoir répondu à la première. Cependant, la valeur de la réponse est importante car la spécialisation est basée sur la réponse spécifique à la question générique. Notons que ce raisonnement demeure valide pour un nombre de questions supérieur à deux.

7.5 Validation de l'aspect transformationnel

Notre travail s'inscrit dans le cadre de l'ingénierie dirigée par les modèles. Dans ce domaine, la validation des transformations est un élément clé. Selon Küster et Abd-El-Razik (2006), prouver que les transformations de modèles sont correctes est une tâche difficile et nécessite des techniques de vérification formelles. En général, la validation de transformation de modèles utilise deux approches : (i) les tests, et (ii) les techniques de vérification. Les tests visent l'exécution de programmes dans des situations représentatives choisies selon des cas de tests appropriés. La validation par les tests est largement appliquée dans l'industrie. Les techniques de vérification visent à analyser complètement la transformation par rapport à certaines propriétés (Küster et Abd-El-Razik, 2006).

Nous avons utilisé l'approche par les tests pour valider nos transformations. Nous avons testé l'ensemble des transformations qui ont servi pour notre référentiel de processus d'affaires. Notons que notre référentiel n'est pas exhaustif car l'ensemble de toutes les combinaisons possibles des réponses aux questions est important. En effet, pour chaque processus, l'ensemble des combinaisons possibles de sept questions est équivalent à l'ensemble de toutes les permutations sans répétition d'un ensemble fini de sept éléments, soit $7!$. Il faut, par la suite, multiplier ce nombre par le nombre de processus et le nombre de réponses possibles (deux dans notre cas). Toutefois, nous pensons que l'ensemble de transformations appliquées peut nous permettre d'expérimenter notre approche, puisque nous avons appliqué toutes les règles de transformation permises pour chaque famille de processus d'affaires du Tableau 7.1.

Pour valider l'aspect transformationnel de notre approche, nous nous sommes principalement intéressés à vérifier la correction de nos transformations. Néanmoins, nous nous sommes aussi penchés sur les questions de confluence des transformations. Pour l'aspect de correction des transformations, nous avons vérifié si les modèles obtenus par l'application de nos règles sont corrects. Il y a deux aspects à considérer dans ce contexte :

1. *La correction syntaxique du modèle obtenu après la transformation* : Cet aspect consiste à vérifier que le modèle transformé est syntaxiquement correct. Pour cela, nous devons nous assurer qu'il est conforme à notre méta-modèle de processus d'affaires (incluant les règles d'affaires).
2. *La correction sémantique du modèle obtenu après la transformation* : Cela consiste à vérifier que le modèle obtenu préserve la sémantique d'affaires du modèle source (i.e. le modèle initial et le modèle final sont sémantiquement équivalents).

7.5.1 La correction syntaxique des transformations

Pour vérifier que les modèles obtenus après les transformations sont conformes à notre méta-modèle de processus d'affaires, nous avons utilisé le cadre de validation d'EMF. Notre technique consistait à vérifier la conformité des modèles intermédiaires et finaux. Nous avons appliqué la validation à chaque vue du processus d'affaires après chaque transformation (i.e. après chaque réponse à une question spécifique) pour vérifier si les règles intermédiaires produisent des modèles valides.

Nous avons appliqué plusieurs transformations de spécialisation aux processus énumérés dans le Tableau 7.1. Nous avons appliqué la même approche de validation utilisée pour vérifier l'aspect de représentation. Les transformations se font de façon correcte (syntaxiquement) même si les instances des modèles se chevauchent. Les modèles obtenus, après application de nos transformations, restent conformes à notre méta-modèle de processus d'affaires.

7.5.2 La correction sémantique des transformations

La validation de la correction sémantique d'une transformation consiste à vérifier si le modèle obtenu après la transformation préserve la sémantique du modèle source. Dans ce contexte, nous cherchons à vérifier que la sémantique REA de chacun des processus de la chaîne de valeur est préservée après l'application des spécialisations reliées aux patrons d'affaires. Toutefois, il n'existe aucune approche pour valider l'équivalence sémantique entre un modèle source et le modèle obtenu après transformation (Varro et Pataricza, 2003). De plus, nous ne pouvons toujours prouver formellement l'équivalence sémantique dans les approches transformationnelles (Varro et Pataricza, 2003). Dès lors, nous avons analysé manuellement les modèles obtenus après l'application de différentes transformations. Pour cela, nous avons présenté notre référentiel¹ contenant un ensemble considérable de modèles sources et de modèles obtenus après les transformations (i.e. modèles finaux après la spécialisation) à nos trois experts en processus d'affaires. Nous voulions savoir si le modèle résultant correspond aux effets escomptés suite aux réponses aux questions posées sur le processus initial. Par exemple, pour la question portant sur l'existence d'un contrat, nous avons déterminé que s'il y avait un contrat entre deux agents A et B, nous devons éliminer les étapes de recherche du partenaire d'affaires. On peut poser la question à l'expert telle quelle « Est-il vrai que lorsqu'il y a un contrat entre deux agents A et B, on doit éliminer les étapes de recherche du partenaire ? ». Cette question étant trop abstraite, plutôt que la poser telle quelle, nous avons présenté aux experts les modèles de processus avant et après transformations et nous leur avons demandé si le processus résultant répondait à leurs attentes.

Pour avoir un avis sur chacune des règles de transformation, nous avons généré les modèles intermédiaires (i.e. le chemin de spécialisation entre les paires de processus directs dans le catalogue se compose d'une séquence contenant un seul élément structurel).

¹Pour faciliter l'interprétation des modèles de processus, nous les avons présentés avec la notation BPMN.

La validation a donné satisfaction pour l'ensemble des transformations de notre catalogue. Notons que nous n'avons pas validé les transformations lorsque les questions ne sont plus valides après certaines spécialisations (voir sect. 7.4.2).

7.5.3 La confluence des transformations

La propriété de confluence exige que l'application des règles de transformation sur un modèle donne toujours le même résultat ou un résultat équivalent quelque soit l'ordre d'application de ces règles (Küster et Abd-El-Razik, 2006). En général, la confluence d'une transformation signifie que le modèle obtenu après l'exécution des règles qui lui sont associées est le même (ou équivalent), indépendamment de l'ordre dans lequel ces règles sont exécutées. Lorsqu'une transformation génère des modèles intermédiaires qui ne peuvent plus être transformés (i.e. problème de terminaison), la transformation n'est pas confluente (Küster et Abd-El-Razik, 2006). La terminaison et la confluence assurent que la transformation produit toujours un modèle unique (Küster, 2006). Cependant, la terminaison des transformations est indécidable selon Heckel, Kuster et Taentzer (2002).

Dans le cadre de notre approche, nous allons vérifier si chaque transformation globale (voir sect. 6.2.1.3) donne le même processus d'affaires indépendamment de l'ordre d'exécution des règles de transformation.

7.5.3.1 La confluence et l'indépendance parallèle

La confluence est souvent reliée au concept d'indépendance parallèle des règles de transformation. Deux règles sont dites parallèlement indépendantes si toutes les applications possibles de ces deux règles ne s'entravent pas les unes les autres. Deux règles de transformations peuvent être considérées parallèlement indépendantes si les éléments communs sur lesquels elles vont s'appliquer, sont préservés par les deux transformations (Corradini *et al.*, 1996). Si deux règles ne sont pas parallèlement indépendantes, alors elles pourront donner lieu à un problème de non-confluence. En effet, dans ce cas, les règles se chevauchent car leurs parties gauches manipulent les mêmes instances d'un modèle.

Dans le cadre de nos transformations, les règles ne sont pas nécessairement parallèlement indépendantes. En effet, nos règles sont basées sur des patrons d'affaires qui

peuvent rentrer en conflit. Prenons l'exemple des règles de transformation associées au patron « *Contrat* » et au patron « *Engagement* ». Ces règles peuvent se chevaucher au niveau des modèles de processus d'affaires dépendamment des réponses aux questions.

Dans la suite de cette section, notons $R_{q4, 1}$ et $R_{q5, 0}$ les règles de transformation correspondantes à une réponse positive à la question Q4 (i.e. question du patron contrat) et une réponse négative à la question Q5 (i.e. question du patron d'engagement), respectivement. Notons aussi $T_g(P, R_{q4, 1}, R_{q5, 0})$, la transformation globale correspondante à l'application des règles $R_{q4, 1}$ et $R_{q5, 0}$ à un processus d'affaires P . Ces deux règles peuvent être en conflit. Par exemple, au niveau de la vue informationnelle, les deux règles se chevauchent au niveau de l'entité qui représente le contrat REA. En effet, la règle $R_{q4, 1}$ doit ajouter des associations entre l'entité du contrat REA et la nouvelle entité qui correspond à l'accord qui régleme les contrats individuels. La règle $R_{q5, 0}$ doit supprimer la structure liée à l'entité contrat. Ce problème est illustré dans les Figures 7.3, 7.4 et 7.5. La Figure 7.3 montre un modèle informationnel générique pour un processus d'échange P avant l'application des règles $R_{q4, 1}$ et $R_{q5, 0}$. La Figure 7.4 montre le modèle obtenu après l'application de la séquence des règles $\langle R_{q4, 1}, R_{q5, 0} \rangle$. La Figure 7.5 montre le modèle obtenu après l'application de la séquence $\langle R_{q5, 0}, R_{q4, 1} \rangle$.

Dans ce qui suit, nous appelons un *ensemble conflictuel* l'ensemble de règles dont les parties gauches manipulent des entités communes. Nous appelons ces entités communes *l'espace conflictuel* dans le contexte de l'ensemble conflictuel. Par exemple, $\{R_{q4, 1}, R_{q5, 0}\}$ est un *ensemble conflictuel* car la partie gauche de ces deux règles manipule l'entité qui correspond au contrat REA. Cette dernière forme *l'espace conflictuel* dans le contexte de l'ensemble conflictuel $\{R_{q4, 1}, R_{q5, 0}\}$. Dans notre approche, un espace conflictuel est formé des concepts d'une vue du processus d'affaires. Par exemple, pour le modèle de la vue informationnelle, l'espace conflictuel peut contenir des entités et des associations, alors que pour la vue dynamique, il peut contenir des séquences et des éléments de contrôles.

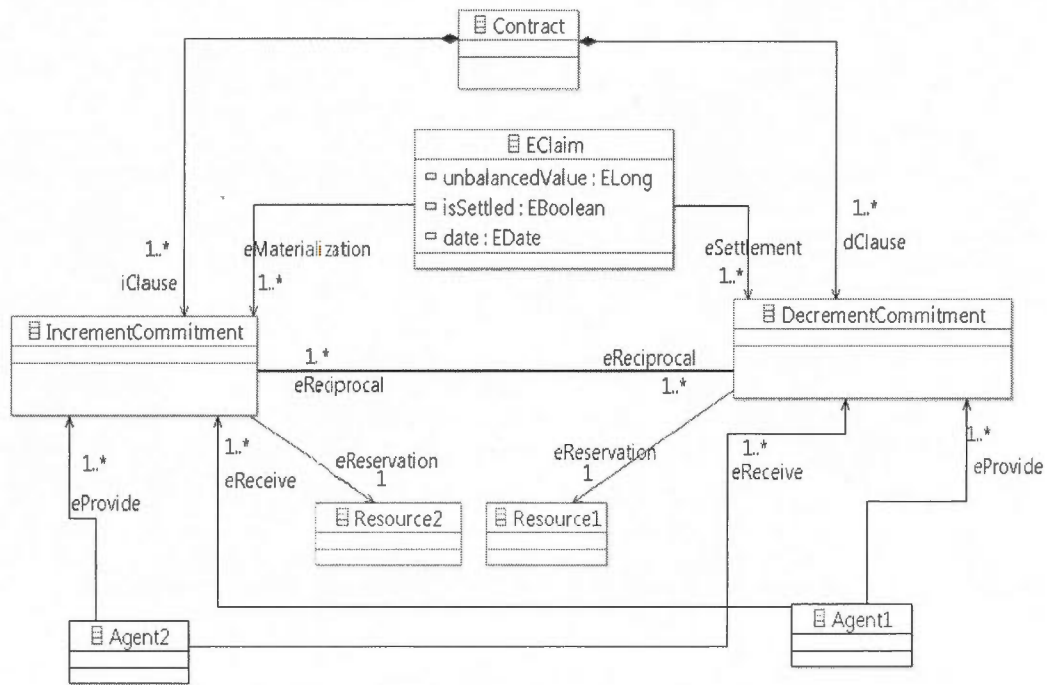


Figure 7.3 La vue informationnelle avant l'application des règles $R_{q4,1}$ et $R_{q5,0}$.

Après l'application de la règle $R_{q4,1}$ (sect. 6.2), nous devons ajouter une nouvelle classe *Agreement* selon le patron « *Contrat* ». Nous devons aussi ajouter des associations bidirectionnelles entre cette nouvelle classe et (i) les agents économiques, et (ii) l'entité qui modélise le contrat REA. Après l'application de la règle $R_{q5,0}$, l'échange devient instantané et les obligations futures ne sont plus nécessaires. Nous devons alors supprimer l'entité qui représente le contrat REA. Par conséquent, l'accord sera supprimé. Aussi, nous devons supprimer la structure de la créance puisque les événements économiques duaux deviennent instantanés en l'absence d'engagements.

Maintenant, nous appliquons la transformation selon la séquence $\langle R_{q5,0}, R_{q4,1} \rangle$. L'application de la règle $R_{q5,0}$ supprime l'entité qui modélise le contrat REA et la structure de la créance. Après l'application de la règle $R_{q4,1}$ (sect. 6.2), nous obtenons le modèle de la Figure 7.5.

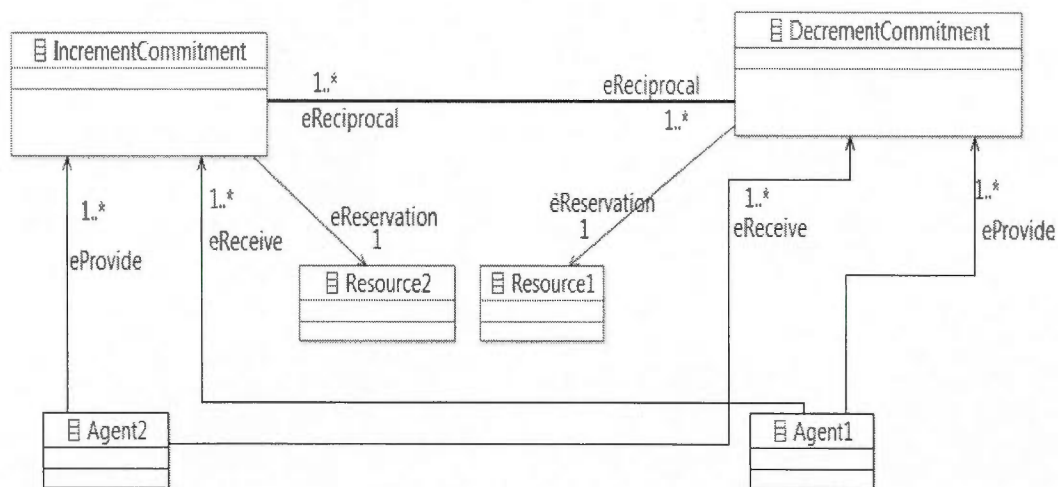


Figure 7.4 La vue informationnelle après l'application des règles $\langle R_{q4, 1}, R_{q5, 0} \rangle$.

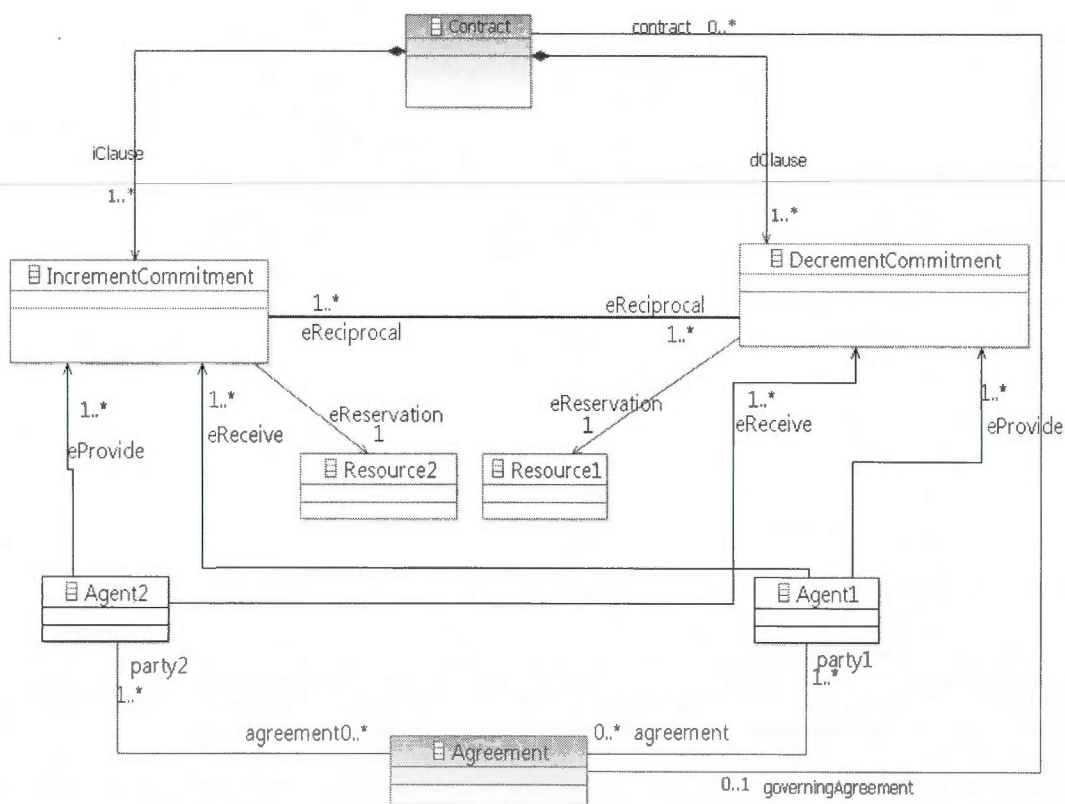


Figure 7.5 La vue informationnelle après l'application des règles $\langle R_{q5, 0}, R_{q4, 1} \rangle$.

Tel qu'illustré dans les Figures 7.4 et 7.5, le modèle obtenu, après l'exécution de la transformation globale $T_g(P, R_{q4, 1}, R_{q5, 0})$ n'est pas toujours le même. En effet, le modèle final dépend de l'ordre d'exécution des deux règles. La propriété de confluence de la transformation $T_g(P, R_{q4, 1}, R_{q5, 0})$ n'est alors pas vérifiée. Dans ce cas précis, nous pouvons déterminer l'ensemble conflictuel et l'espace conflictuel au moment de la conception des règles. Nous pouvons aussi remarquer que les deux règles $R_{q4, 1}$ et $R_{q5, 0}$ sont mutuellement exclusives. En effet, l'application de ces deux règles dans le même processus de spécialisation est sémantiquement fausse. On ne peut pas dire qu'on veuille un contrat ($R_{q4, 1}$) et, qu'en même temps, on ne veuille pas modéliser les engagements ($R_{q5, 0}$) car un contrat régleme les engagements (Hruby, 2006). Lorsque deux règles sont mutuellement exclusives, il faut prévoir l'application d'une seule règle dans le processus de spécialisation. Si deux règles ne sont pas mutuellement exclusives et qu'elles posent un problème de confluence, il faut imposer un ordre d'exécution.

Devant cette problématique de confluence, nous pensons qu'il est possible de proposer un support à l'analyste lors du processus de spécialisation. Ce support, basé sur les réponses aux questions, doit permettre à l'analyste de voir la dépendance entre les règles afin pouvoir définir éventuellement un ordre d'application des questions/réponses. Cet ordre nous permettra d'établir la séquence d'exécution des règles de transformation.

7.6 Conclusion

L'expérimentation, présentée dans ce chapitre, nous a permis de vérifier les trois aspects principaux de notre approche de spécification de processus d'affaires : (1) la représentation de processus d'affaires, (2) l'applicabilité des questions génériques, et (3) les transformations de spécialisation.

La validation de l'aspect de la représentation de processus d'affaires montre que notre méta-modèle a permis de modéliser un ensemble raisonnable de processus d'affaires provenant de différents domaines.

Pour l'aspect d'applicabilité des questions génériques, comme les résultats des tests l'ont démontré, les questions spécifiques, obtenues après instanciation des questions

génériques, s'appliquent bien à différents processus d'affaires. De plus, elles ont préservé leur sémantique d'affaires dans le contexte de l'ensemble de processus cibles quand celles-ci étaient applicables.

Pour l'aspect transformation, nous nous sommes penchés essentiellement sur les questions de correction syntaxique, de correction sémantique et de confluence. L'expérimentation était satisfaisante. En effet, le cadre de validation d'EMF nous a permis de valider la correction syntaxique des modèles obtenus après transformation et a donné satisfaction dans 100% des cas testés. La validation par des experts de l'équivalence sémantique entre les modèles source et ceux obtenus après les transformations a donné satisfaction dans l'ensemble des cas validés (i.e. l'ensemble des processus d'affaires de notre catalogue).

Dans le cadre de la confluence des transformations (i.e. l'application des règles de transformation sur un modèle donne toujours le même résultat), nos tests ont démontré que nos transformations peuvent poser un problème de confluence lorsque les règles -qui sont associées aux patrons d'affaires- se chevauchent. Dans ce cas, il faut imposer un ordre d'exécution des règles au préalable.

CONCLUSION ET PERSPECTIVES

Dans cette thèse, nous avons traité de la réutilisation et l'adaptation de processus d'affaires. Nous avons présenté, tout au long de ce travail, une nouvelle méthodologie et des outils qui permettent à un analyste d'affaires de spécifier facilement un nouveau processus d'affaires qui répond à certains besoins d'affaires à partir d'un autre processus générique. Afin de conclure cette thèse, nous allons d'abord résumer notre approche. Par la suite, nous présentons une synthèse de nos contributions, les améliorations et les travaux en perspective.

Résumé

Nous avons présenté dans ce travail une nouvelle approche de spécification de processus d'affaires basée sur les questions. Nous avons montré que cette spécification passe par : (i) une représentation précise de processus d'affaires, (ii) une organisation hybride combinant une classification descriptive et structurelle, et (iii) un ensemble de techniques de spécialisation permettant de passer d'un modèle générique de processus d'affaires vers un modèle adapté aux besoins spécifiques des entreprises.

Notre approche a été présentée dans le chapitre 3. Nous avons commencé par présenter une première démarche de spécification de processus d'affaires dite orientée *métalangage* (voir chap. 3). Cette dernière se base sur un méta-modèle traditionnel de processus d'affaires. Cette approche suit les étapes suivantes : 1) à partir de questions spécifiques, 2) on les généralise via le méta-modèle de processus d'affaires, et 3) on développe des opérateurs de spécialisation basés sur les questions génériques et les réponses spécifiques. La spécialisation suit un processus similaire à l'approche ontologique (i.e. on instancie les questions dans le contexte d'un processus en entrée et on applique, par la suite, les opérateurs de spécialisation selon les réponses aux questions spécifiques).

Une validation préliminaire a montré que les questions génériques obtenues après la généralisation des questions spécifiques ont perdu leur sémantique d'affaires initiale. Il en résultait deux inconvénients majeurs: (i) l'instanciation des questions génériques générait beaucoup de questions superflues, et (ii) il était très difficile de trouver des opérateurs de spécialisation génériques qui peuvent s'appliquer à différents groupes de processus d'affaires.

Suite aux problèmes rencontrés dans la première approche, nous avons cherché à abstraire les processus d'affaires spécifiques et leurs variantes sur le plan métier pour obtenir des questions génériques et des opérateurs de spécialisation qui préservent la sémantique d'affaires et qui s'appliquent à des processus appartenant à différents domaines. Par conséquent, nous avons cherché une ontologie d'affaires qui nous permet de voir les similitudes entre les différents processus d'affaires. Dès lors, nous avons proposé d'utiliser l'ontologie d'affaires REA. L'ontologie REA, présentée dans le chapitre 4, perçoit les différentes activités/processus d'affaires comme des *échanges* ou des *conversions* de *ressources économiques* effectuées par des *agents*. À partir de là, il est possible de percevoir un processus d'achat et un processus d'embauche comme essentiellement des *échanges*, dans un cas d'argent contre un produit, et dans l'autre, d'argent contre de la main-d'œuvre. Des questions portant sur les *termes* de cet échange peuvent alors s'appliquer autant à l'approvisionnement qu'à la gestion de ressources humaines. Il en est de même pour les opérateurs de spécialisation. L'*échange* et la *conversion* sont deux *patrons REA* de base, desquels découlent d'autres patrons spécialisés. Ces autres patrons offrent, à leur tour, des variantes, i.e. des *questions* et des *opérateurs de spécialisation*. Notre nouvelle approche de représentation de processus d'affaires, *orientée patrons d'affaires REA*, a été présentée dans le chapitre 5.

L'approche de *spécialisation* de processus d'affaires que nous proposons a été présentée dans le chapitre 6. Cette dernière est basée sur les questions génériques portant sur les patrons REA—et donc avec un sens métier—et des transformations de spécialisation qui s'appliquent à des processus d'affaires provenant de différents domaines.

Pour prouver la faisabilité de notre méthodologie, nous avons implémenté les différents concepts de l'approche. Nous avons mis en œuvre notre méta-modèle de processus d'affaires avec le cadre de modélisation EMF. Nous avons dû augmenter le méta-modèle

REA de base pour supporter les événements d'affaires et informationnels. Nous avons aussi augmenté le méta-modèle de l'ontologie REA pour y ajouter le concept d'orchestration en se basant sur le modèle comportemental de la norme BPDm (OMG, 2008). Pour implémenter nos transformations de spécialisation, nous avons utilisé la technologie des moteurs à base de règles. Nous avons implémenté un ensemble de règles de production avec le système de gestion de règles d'affaires Drools (JBoss, 2012).

Pour valider notre approche, nous avons souhaité nous assurer que : 1) l'on peut représenter plusieurs processus d'affaires provenant de différents domaines, 2) nos questions génériques peuvent être appliquées à différents processus d'affaires indépendamment de leurs domaines, et 3) nos transformations de spécialisation sont correctes (i.e. génèrent des modèles syntaxiquement et sémantiquement corrects). Nos données expérimentales et la validation sont décrites dans le chapitre 7. Sommairement, nous avons montré que : 1) les résultats pour l'aspect de représentation sont positifs, 2) les questions s'appliquent généralement bien, tout en préservant leur sémantique, et 3) la validation des transformations a donné satisfaction dans 100% des cas testés. Cependant, nos tests ont démontré que nos transformations posent un problème de confluence lorsque les règles qui sont associées aux patrons d'affaires se chevauchent.

Contributions

Nous avons proposé dans cette thèse une nouvelle méthodologie de réutilisation et d'adaptation de processus d'affaires par les questions. Notre approche se caractérise principalement par son niveau d'abstraction ontologique qui l'a rendue générique. C'est grâce à cet aspect de généricité que nous étions capables d'apporter notre contribution à cinq niveaux que nous énumérons ci-dessous :

1. Génération sur demande de processus spécifiques

Aucune approche de modélisation de processus d'affaires orientée catalogue ne peut prétendre être complète/exhaustive (i.e. couvrir *tous* les besoins) et ce pour deux raisons : 1) il n'est pas pratique de recenser *tous* les processus d'affaires, mais aussi, 2) de nouvelles pratiques d'affaires émergent sans cesse, donnant lieu à de nouveaux processus et de

nouvelles variantes de processus existants. Par exemple, avec le manuel MIT (Malone *et al.* 1999), nous devons identifier et codifier manuellement toutes les variantes.

Notre **première contribution** consiste en la **génération dynamique/automatique de variantes de processus d'affaires**. Grâce aux procédures génériques que nous avons proposées, nous n'avons pas besoin d'un catalogue exhaustif de processus d'affaires. En effet, notre spécialisation permet de générer des processus spécifiques dynamiquement en se basant seulement sur les questions que notre outil pose à l'analyste. De plus, notre approche de spécification ne dépend pas du domaine d'affaires, contrairement aux approches proposées par Fowler (1997), Eriksson et Penker (2000), et Wohed (2000). Aussi, elle n'exige pas la connaissance préalable des variantes des processus d'affaires comme c'est le cas du cadre Provop (Hallerbach, Bauer et Reichert, 2008).

2. Une organisation structurelle de processus d'affaires

La deuxième nouveauté de notre approche consiste en la construction d'un catalogue de processus génériques basé sur une classification *structurelle*. En effet, notre classification utilise la structure de notre méta-modèle de processus d'affaires, plus précisément, le concept de < question, réponse > (i.e. l'élément structurel). Cette structure permet de construire une hiérarchie de spécialisation où chaque élément hérite des caractéristiques de son parent. Ce type de catalogue favorise la compréhensibilité, la maintenabilité, et la réutilisabilité (Wyner et Jintae, 2003). De plus, il permet la génération de modèles alternatifs tout en offrant une organisation qui permet d'indexer et de localiser facilement les processus. Bien que certaines initiatives de spécification de processus d'affaires utilisent une approche de catalogage (e.g. Malone *et al.*, 1999; Lefebvre et De Lucca, 1999; Adams *et al.*, 2009), aucune d'elles n'offre une hiérarchie *structurelle*.

3. Une spécialisation par des questions génériques avec un sens métier

À notre connaissance, aucune initiative de spécification de processus d'affaires par les questions n'a pu offrir des questions indépendantes des domaines d'affaires (i.e. génériques) et qui ont une sémantique métier permettant à des analystes d'affaires d'appliquer la méthode facilement.

Comme nous l'avons présenté dans le chapitre 6, notre approche se base sur des questions génériques qui s'appliquent à des processus d'affaires provenant de différents domaines. En outre, nos questions spécifiques, obtenues après l'instanciation des questions génériques, conservent un sens métier qui apporte un grand avantage quant à l'application de l'approche. En effet, c'est le sens métier de ces questions qui permet à un utilisateur disposant seulement de connaissances liées au domaine d'affaires, d'appliquer notre approche; celle-ci se basant uniquement sur les réponses aux questions pour générer de nouveaux processus d'affaires. Cela est rendu possible grâce à l'abstraction par l'ontologie d'affaires REA. Concrètement, nos questions génériques se basent sur la sémantique de patrons d'affaires REA. C'est cette sémantique qui nous a permis de définir directement (i.e. sans passer par une étape de généralisation) des questions génériques qui portent un sens métier.

4. Des opérateurs de spécialisation génériques et faciles à appliquer

Notre quatrième contribution consiste en la proposition d'opérateurs de spécialisation génériques. En effet, nos transformations de spécialisation ne dépendent pas de domaines d'affaires particuliers. Ils permettent d'adapter des processus d'affaires provenant de différents domaines. Tout comme nos questions génériques, cela a été possible grâce à l'abstraction offerte par l'ontologie REA. À notre connaissance, les approches de spécialisation de processus d'affaires existantes posent plusieurs problèmes. En effet, soit (i) elles sont spécifiques à des processus particuliers, soit (ii) elles sont trop génériques pour être appliquées, ou (iii) elles exigent la connaissance préalable des variantes possibles, ou encore, (iv) elles sont complexes (voir chap. 2).

5. Une preuve de faisabilité de l'approche

Sur le plan pratique, nous avons développé une application qui permet à un analyste d'affaires de choisir un processus d'affaires et de le spécialiser. Essentiellement, notre application est basée sur les questions, les modèles de processus d'affaires et les règles de transformation. Notre application inclut :

- *Un méta-modèle pour représenter les processus d'affaires* : notre méta-modèle de processus d'affaires est basé sur l'ontologie d'affaires REA à laquelle nous avons

ajouté le modèle comportemental de la norme BPDM (OMG, 2008). Nous avons implémenté ce méta-modèle avec le cadre de modélisation EMF.

- *Des règles de transformation pour la spécialisation des processus d'affaires* : nous avons représenté les transformations inhérentes à la spécialisation des vues par un ensemble de règles de production que nous avons implémentées avec la technologie des moteurs d'inférence, en l'occurrence, le système de gestion de règles d'affaires Drools (JBoss, 2012).
- *Un méta-modèle pour représenter les questions génériques* : après examen de différents patrons d'affaires REA, nous avons construit un langage pour décrire les questions génériques. Nous avons implémenté ce langage avec un schéma XML.

Travaux futurs

L'approche présentée dans cette thèse s'est avérée prometteuse. Cependant, les expérimentations que nous avons menées (voir chap. 7) ont soulevé certains problèmes qu'il faudra traiter pour rendre notre approche plus complète.

1. Exploitation non monotone des transformations

Ce point est relié au mode d'exploitation *non monotone* des approches transformationnelles. Nous avons montré dans le chapitre 6 que notre approche transformationnelle exploite le mode monotone seulement. Dans ce mode de fonctionnement, nous ne considérons pas les modèles intermédiaires après chaque transformation. Or, ces modèles peuvent affecter le processus de spécialisation. En effet, si ces modèles sont ajoutés à la base de faits, ils peuvent provoquer des changements au niveau des règles (e.g. le déclenchement de nouvelles règles). Par exemple, après la sous-traitance d'un processus de la chaîne de valeur, un nouveau processus d'échange (i.e. achat de service) est créé (voir le patron de sous-traitance au chapitre 6). Dans ce contexte, il est important de considérer ce nouveau processus d'affaires dans la spécialisation.

2. Gestion des transformations conflictuelles

Comme nous l'avons expliqué dans la section 7.5 du chapitre 7, nous avons rencontré certains problèmes bien connus avec les approches de transformation de modèles. Notamment, le problème de non-confluence de transformations lorsque différentes règles s'appliquent au même fragment d'un modèle. Nous avons présenté un exemple qui montre la non-confluence de deux règles de transformation. Cela pose problème car le processus d'affaires obtenu après la spécialisation (transformation finale) peut ne pas répondre aux besoins des analystes. Dans ce contexte, nous pouvons explorer deux pistes pour résoudre ce problème. Une première piste consiste à construire les paires critiques au moment de la conception (Küster, 2006). L'idée des paires critiques est, dans un premier lieu, de capturer les étapes des règles problématiques dans un *contexte minimal*. Ce contexte est dit *minimal* car, il se base sur quelques étapes de règles et non la règle au complet (i.e. une section critique de la partie gauche de la règle et non la partie gauche au complet). Par la suite, il faut analyser si nous pouvons dériver un modèle successeur commun pour les règles chevauchantes. Une deuxième piste consiste à appliquer les règles de transformation en parallèle et, par la suite, utiliser des règles pour fusionner les modèles résultant de ces transformations. Dans cette perspective, nous pouvons nous inspirer des règles de fusion de fragments de modèles développées dans les approches transformationnelles en orienté objet. Par exemple Marschall et Braun (2003) proposent de fusionner les objets ayant la même identité en un seul objet et les associations de même type entre deux objets ayant les mêmes identités en une seule association.

3. Couverture des patrons d'affaires

Comme nous l'avons expliqué dans les chapitres 3 et 6, notre approche de spécification de processus d'affaires est basée sur les patrons d'affaires REA. En effet, c'est à partir de ces patrons que nous avons identifié les variantes possibles de processus d'affaires. Cela soulève la problématique de *couverture* des patrons d'affaires, que l'on peut poser ainsi : 1) avons-nous toutes les questions et tous les opérateurs de spécialisation dont nous pourrions avoir besoin pour notre approche ?, et 2) comment nous en assurer si la réponse est non ? La réponse à la première question est, naturellement, négative. Tout ce que nous pouvons faire, c'est d'assurer une *meilleure* couverture en continuant à recenser et à codifier des patrons

d'affaires REA. Toutefois, nous pensons que les patrons d'affaires les *plus utiles* ont probablement déjà été identifiés et, tout comme pour les patrons de conception, les nouveaux patrons qui pourront être recensés dans la littérature couvriront, forcément, des besoins de plus en plus restreints.

Pour découvrir de nouveaux patrons d'affaires, Hruby (2006) propose l'application d'un patron basé sur le concept de *Inventer's Paradox* du mathématicien George Polya (1982). Pour résoudre un problème (pas nécessairement mathématique), Polya préconise l'utilisation de certaines heuristiques. Essentiellement, pour résoudre un problème spécifique, Polya propose d'essayer de résoudre le problème générique. La résolution du problème générique est paradoxalement plus simple. Pour cela, il faut trouver un moyen approprié pour généraliser le problème spécifique. Dans le contexte de nos patrons d'affaires, cette démarche se traduit tout d'abord par l'analyse du problème d'affaires spécifique de l'organisation. Par la suite, il faut essayer de le généraliser, si possible, sous forme d'un ou plusieurs patrons d'affaires. Cette généralisation, doit être indépendante du domaine d'affaires du problème spécifique afin d'offrir une solution aux problèmes spécifiques initiaux.

4. Optimiser la recherche du processus le plus proche

Nous avons expliqué, dans le chapitre 5, que la recherche d'un processus d'affaires se base sur le chemin de spécialisation. Nous avons mentionné, dans le chapitre 6, que nous ne générons pas les modèles intermédiaires dans le catalogue. Nous avons montré que cela pose problème lors de la recherche du processus générique le plus proche des besoins de l'organisation.

Par exemple, dans le cas où le processus d'affaires recherché n'existe pas, nous essayons de trouver le processus le plus proche avec la séquence optimale du chemin de spécialisation à partir de la racine d'une famille de processus. Cette séquence de (question, réponse) doit être idéalement maximale (i.e. contient le maximum de réponses aux questions de l'analyste dans le bon ordre). Supposons que nous cherchions le processus du catalogue le plus proche réalisant la séquence $c = \langle (Qa, Ra), (Qb, Rb), (Qc, Rc), (Qd, Rd) \rangle$. Supposons que dans le catalogue, nous ayons les deux processus $p1$ et $p2$ obtenus, à partir du même processus, par les chemins de spécialisation $c1 = \langle (Qa, Ra) \rangle$ et $c2 = \langle (Qa, Ra), (Qb, Rb), (Qc, Rc), (Qd, Rd), (Qt, Rt) \rangle$, respectivement. Comme

nous ne gardons pas les modèles intermédiaires, notre technique actuelle choisira le processus $p1$ et non le processus $p2$. Pourtant, il est plus intéressant de choisir $p2$ en éliminant (Qt, Rt) , car le nombre de questions/réponses appliquées dans l'ordre pour obtenir $P2$ est plus intéressant. En effet, si nous définissons les opérations d'intersection (\cap) et de cardinalité (ω) entre les séquences qui représentent les chemins de spécialisation, nous obtenons l'inégalité : $\omega(c \cap c2) > \omega(c \cap c1)$. Pourtant, notre approche choisira la séquence $c1$ et non $c2$.

Nous voyons deux approches possibles pour améliorer la recherche. Soit nous devons sauvegarder les modèles intermédiaires ou bien nous devons permettre d'inverser des transformations. Dans notre cas, nous devons, par exemple, inverser la transformation liée à l'application de la réponse Rt à la question Qt .

5. Difficulté du passage du modèle BPMN vers REA et inversement

Nous avons montré, dans les chapitres 3, 5, 6 et 7, que notre méthodologie préconise le passage d'une représentation de processus d'affaires de la norme BPMN vers une représentation basée sur REA et inversement. Cependant, cela soulève certains problèmes que nous avons rencontrés lors de l'expérimentation de notre approche. En effet, nous avons remarqué, par exemple, que les événements REA duaux pour les processus de conversion ne sont pas toujours présents dans le processus BPMN. Aussi, la distinction entre les différents types d'événements n'est pas intuitive. Finalement, le passage inverse de notre représentation vers BPMN n'est pas possible sans la recomposition des vues.

Comme piste pour permettre d'automatiser notre méthodologie de conception du processus générique par un passage automatique d'un modèle de processus en BPMN à un modèle de processus décomposé en vues de processus d'affaires (i.e. vues REA, dynamique, organisationnelle et informationnelle), nous proposons d'identifier les différents types d'événements REA (e.g. par annotation). De plus, il faut s'assurer, pour chaque processus de conversion de la chaîne de valeur, que les événements duaux sont présents.

Une solution possible pour la recomposition consiste à utiliser un processus similaire à celui proposé par Davids (1997) pour l'intégration des vues REA. D'abord, Davids (1997) propose de relier les différents processus REA en se basant sur leurs entités

communes. Par la suite, l'analyste doit effectuer une autre étape pour s'assurer de la complétude du diagramme obtenu. Cette dernière étape comporte deux vérifications. La première consiste à vérifier que chaque ressource économique est reliée à au moins un événement d'incrément ou à au moins un événement de décrétement. La deuxième validation consiste à vérifier que chaque événement économique est relié à au moins un autre événement économique par le biais de la relation de dualité d'échange ou de conversion.

Conclusion

Notre travail s'inscrit dans le cadre plus général de la conception d'une méthode de développement de logiciels à partir de la description de besoins d'affaires, en l'occurrence, à partir de modèles de processus d'affaires. Cela a été un rêve de chercheurs et de praticiens depuis les années 70. Or, une telle approche repose sur la capacité des analystes d'affaires à fournir des modèles de processus *détaillés et fidèles*. En offrant aux analystes la possibilité de générer automatiquement de tels modèles, à partir de catalogues de processus génériques, nous espérons contribuer à nous rapprocher de l'objectif de développement de systèmes d'information à partir de modèles de processus d'affaires. Notre approche s'est avérée prometteuse. Pour une mise en œuvre robuste, plusieurs défis de recherche, certains communs aux approches transformationnelles et d'autres spécifiques à notre approche, devront être relevés.

APPENDICE A

LES QUESTIONS DE SPÉCIALISATION DE WOHEd

Voici les questions relatives au domaine de la réservation qui ont été proposées par Petia Wohed dans son travail initial (Wohed, 2000) :

1. Est-ce que la réservation consiste en un ou plusieurs objets ?
2. Est-ce que la réservation concerne un objet spécifique ou les caractéristiques d'un objet ?
3. Est-ce que toutes les réservations ont le même caractère ou sont-elles divisées en plusieurs catégories ?
4. Est-ce qu'il est nécessaire de conserver les informations sur la requête de réservation avant de procéder à la réservation ?
5. Est-ce que la réservation nécessite un motif ?
6. Est-ce que la réservation peut être faite par l'intermédiaire d'une partie tierce ?
Si oui, est-ce qu'il est important de mémoriser les informations sur cette partie ?

Ces mêmes questions ont été généralisées dans un deuxième travail en 2001 (Wohed, 2001). Cette généralisation a été réalisée en liant ces questions (spécifiques au domaine d'affaires de la réservation) aux patrons *Transaction* de Coad, North et Mayfield (1997) (voir sect. 2.3.1). Voici les questions génériques obtenues:

1. Est-ce que la transaction consiste en un ou plusieurs objets ?

2. Est-ce que la transaction concerne un objet spécifique ou les caractéristiques d'un objet ?
3. Est-ce que toutes les transactions ont le même caractère ou sont-elles divisées en plusieurs catégories ?
4. Est-ce qu'il est nécessaire de conserver les informations sur la transaction qui devance la transaction en cours? Si oui, est ce que la transaction qui précède concerne un (des) objet(s) spécifique(s) ou la spécification de(s) objet(s) ?
5. Est-ce que la transaction doit être associée à d'autres éléments ?
6. Est-ce que la transaction est faite par l'intermédiaire d'une partie tierce ? Si oui, est-ce qu'il est important de mémoriser les informations sur cette partie ?

APPENDICE B

LE MÉTA-MODÈLE EMF

Le diagramme de la Figure B.1 montre le méta-modèle du cadre EMF. La classe `EPackage` regroupe un ensemble de classes ou de types de données (`EClassifier`). Une instance de `EPackage` représente un paquetage qui correspond à un modèle ecore (ou un fragment de modèle). Une instance de `EPackage` a un identificateur unique. Cet identificateur est défini par l'attribut `nsURI` qui correspond à l'espace de nom associé à cette instance.

Un attribut (`EAttribute`) et une référence (`EReference`) forment les éléments structuraux d'une classe EMF. Ils ont plusieurs propriétés communes regroupées dans la classe `EStructuralFeature`. Cette dernière est une sous-classe de la classe `ETypedElement` qui regroupe les propriétés communes aux éléments du modèle qui sont typés comme les attributs `EAttribute`, les références `EReference`, les paramètres `EParameter` et les opérations `EOperation`. Un type d'une entité typée est défini par la référence `eType` qui est de type `EClassifier`. Cette dernière peut être de type `EClass` et `EDataType` permettant ainsi à un type d'être une classe ou un type de données. Pour ce qui est des types de données, le cadre EMF réutilise les types définis en Java.

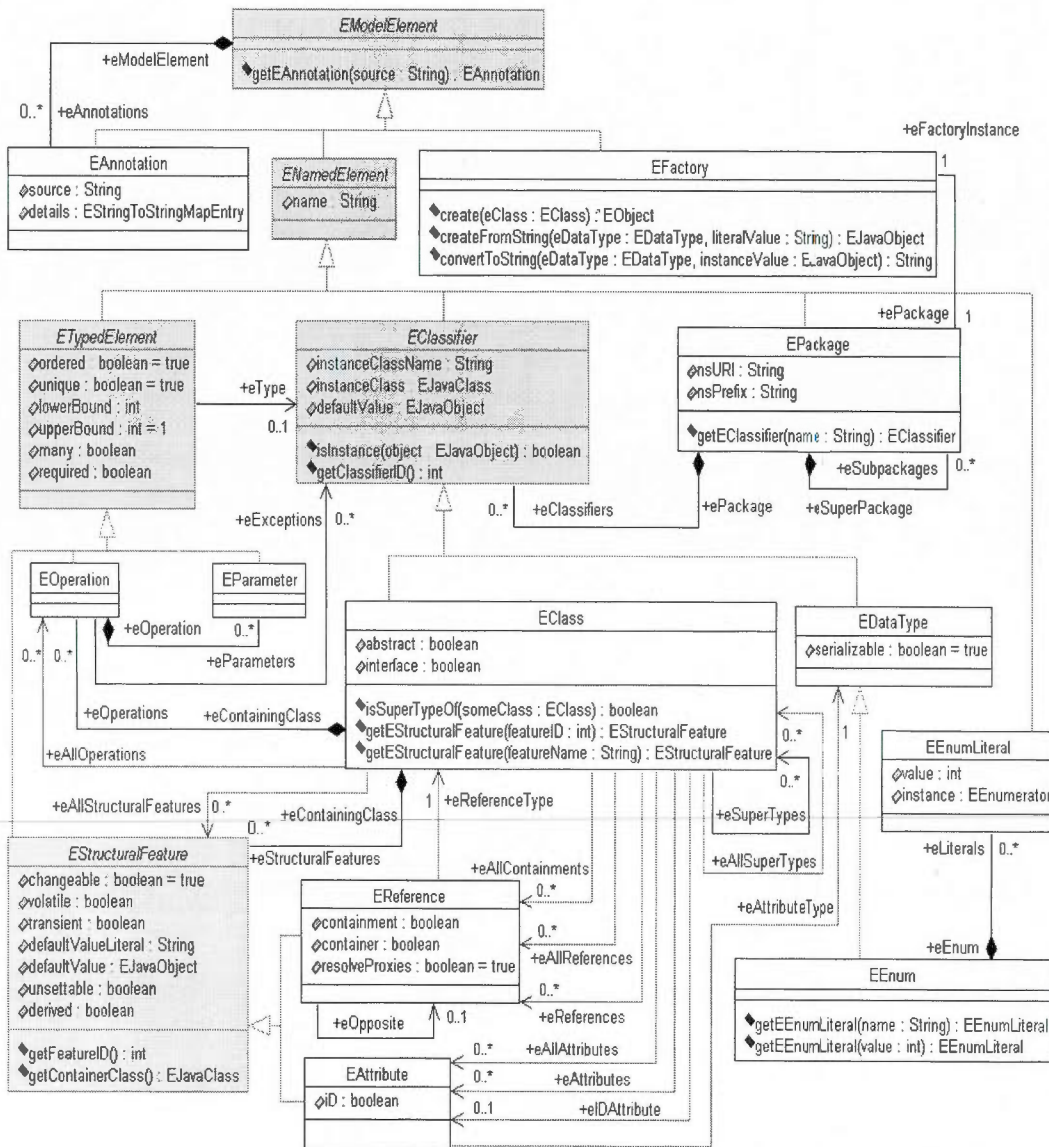


Figure B.1 Le méta-modèle EMF. (Source : Eclipse, 2008)

APPENDICE C

NOTRE MÉTA-MODÈLE DE PROCESSUS D'AFFAIRES

Le diagramme de la Figure C.1 montre l'essentiel de notre méta-modèle de processus d'affaires. Ce méta-modèle est mis en œuvre avec le cadre de modélisation EMF (voir sect. 5.2). Comme nous l'avons mentionné, notre méta-modèle de processus d'affaires se base sur le méta-modèle de l'ontologie d'affaires REA que nous avons augmenté pour : (i) ajouter les événements d'affaires et informationnels qui nous permettent de décrire les processus d'affaires de nos jours, et (ii) ajouter une partie du méta-modèle comportemental (*Behavior model*) de BPDm pour décrire la vue dynamique des processus d'affaires.

Les règles suivantes ont été mises en œuvre avec le Framework de validation EMF :

1. Chaque événement d'incrément doit être relié à au moins un événement de décrétement par la relation de dualité (*dual*).
2. Chaque événement de décrétement doit être relié à au moins un événement d'incrément par la relation de dualité (*dual*).
3. Pour chaque événement économique, l'agent *agentProvide* est différent de l'agent *agentReceive*.
4. Chaque engagement d'incrément doit être relié à au moins un engagement de décrétement par la relation de réciprocité (*reciprocal*).

5. Chaque engagement de décrémentation doit être relié à au moins un engagement d'incrément par la relation de réciprocité (*reciprocal*).
6. Chaque engagement d'incrément doit être relié à au moins un événement économique d'incrément par la relation *economicEvent*.
7. Chaque engagement de décrémentation doit être relié à au moins un événement économique de décrémentation par la relation *economicEvent*.

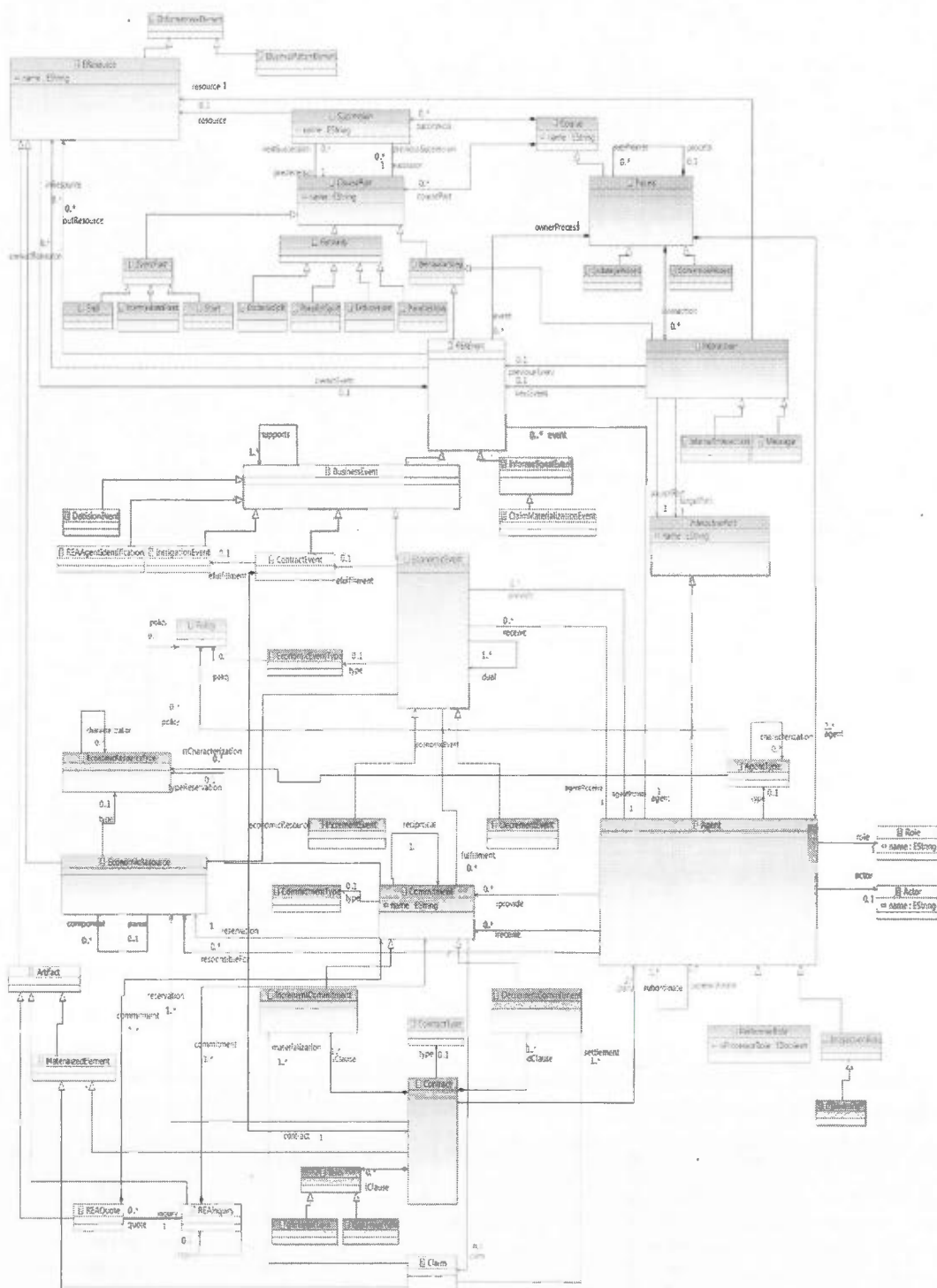


Figure C.1 Le diagramme EMF de notre méta modèle de processus d'affaires.

APPENDICE D

LA VUE REA DU PROCESSUS DE VENTE ET DISTRIBUTION

Dans le chapitre 5, nous avons présenté la vue REA du processus de vente et distribution sous forme d'un composant JFace sous Eclipse (fig. 5.20). Cette représentation graphique ne montrait pas toutes les propriétés des différents éléments de la vue. Voici la vue REA complète sous format XMI.

```
<bp:ExchangeProcess xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:bp="http://www.latece.uqam.ca/bp/1.0"
xsi:schemaLocation="http://www.latece.uqam.ca/bp/1.0 ../bpsf.modeling/model/bp.ecore"
name="SD">

<event xsi:type="bp:REAAgentIdentification" name="RFQ Event"
  outResource="//@event[name='RFQ%20Event']/@ownedResource[name='RFQ']" >
  <ownedResource xsi:type="bp:Artifact" name="RFQ"/>
  <agent href="OrganizationalView.xmi#//@agent[name='Customer']"/>
</event>

<event xsi:type="bp:REAAgentIdentification" name="Quotation Event"
  inResource="//@event[name='RFQ%20Event']/@ownedResource[name='RFQ']"
  outResource="//@event[name='Quotation%20Event']/@ownedResource[name='Quotation']" >
```

```

<ownedResource xsi:type="bp:Artifact" name="Quotation"/>

<agent href="OrganizationalView.xmi#//@agent[name='Sales%20Department']"/>

</event>

<event xsi:type="bp:BusinessEvent" name="Process Purchase"
  inResource="//@event[name='Quotation%20Event']/@ownedResource[name='Quotation']"
  outResource="//@event[name='Process%20Purchase']/@ownedResource[name='PO']">

  <ownedResource xsi:type="bp:Contract" name="PO">

    <party href="OrganizationalView.xmi#//@agent[name='Sales%20Department']"/>

    <party href="OrganizationalView.xmi#//@agent[name='Customer']"/>

  </ownedResource>

  <agent href="OrganizationalView.xmi#//@agent[name='Customer']"/>

</event>

<event xsi:type="bp:ContractEvent" name="SO Event"
  inResource="//@event[name='Process%20Purchase']/@ownedResource[name='PO']"
  outResource="//@event[name='SO%20Event']/@ownedResource[name='SO']">

  <ownedResource xsi:type="bp:Contract" name="SO">

    <party href="OrganizationalView.xmi#//@agent[name='Sales%20Department']"/>

    <party href="OrganizationalView.xmi#//@agent[name='Customer']"/>

  </ownedResource>

  <agent href="OrganizationalView.xmi#//@agent[name='Sales%20Department']"/>

</event>

<event xsi:type="bp:DecrementEvent" name="Sales"
  inResource="//@event[name='SO%20Event']/@ownedResource[name='SO']"
  outResource="//@event[name='Sales']/@ownedResource[name='Product']"
  economicResource="//@event[name='Sales']/@ownedResource[name='Product']">

  <ownedResource xsi:type="bp:EconomicResource" name="Product"/>

  <agent href="OrganizationalView.xmi#//@agent[name='Sales%20Department']"/>

```



```

</event>

<event xsi:type="bp:ContractEvent" name="DO Event"
  outResource="//@event[name='DO%20Event']/@ownedResource[name='DO']">
  <ownedResource xsi:type="bp:Contract" name="DO">
    <party href="OrganizationalView.xmi#@agent[name='Sales%20Department']"/>
    <party href="OrganizationalView.xmi#@agent[name='Distribution%20Department']"/>
  </ownedResource>
  <agent href="OrganizationalView.xmi#@agent[name='Sales%20Department']"/>
</event>

<event xsi:type="bp:ClaimMaterializationEvent" name="Materialize Claim"
  outResource="//@event[name='Materialize%20Claim']/@ownedResource[name='Materialized%20Claim']">
  <ownedResource xsi:type="bp:MaterializedElement" name="Materialized Claim">
    <agent href="OrganizationalView.xmi#@agent[name='Sales%20Department']"/>
  </event>

<event xsi:type="bp:BusinessEvent" name="Process Claim Sales"
  inResource="//@event[name='Materialize%20Claim']/@ownedResource[name='Materialized%20Claim']">
  <agent href="OrganizationalView.xmi#@agent[name='Customer']"/>
</event>

<event xsi:type="bp:IncrementEvent" name="Payment Receipt"
  inResource="//@event[name='Cash%20Disbursement']/@ownedResource[name='Cash']"
  economicResource="//@event[name='Cash%20Disbursement']/@ownedResource[name='Cash']"
  dual="//@event[name='Sales']">
  <agent href="OrganizationalView.xmi#@agent[name='Sales%20Department']"/>
</event>

<subProcess xsi:type="bp:ConversionProcess" name="Distribution">

```

```

<event xsi:type="bp:IncrementEvent" name="Distribution"
  inResource="//@event[name='DO%20Event']/@ownedResource[name='DO']"
  outResource="//@event[name='Sales']/@ownedResource[name='Product']"
  economicResource="//@event[name='Sales']/@ownedResource[name='Product']"
  dual="//@subProcess[name='Distribution']/@event[name='Product%20Use']">

  <ownedResource xsi:type="bp:Artifact" name="DR"/>

  <agent
    href="OrganizationalView.xmi#//@subProcess[name='Distribution']/@agent[name='Distributi
on%20Department']"/>

  </event>

<event xsi:type="bp:DecrementEvent" name="Product Use"
  inResource="//@event[name='Sales']/@ownedResource[name='Product']"
  economicResource="//@event[name='Sales']/@ownedResource[name='Product']"
  dual="//@subProcess[name='Distribution']/@event[name='Distribution']">

  <agent
    href="OrganizationalView.xmi#//@subProcess[name='Distribution']/@agent[name='Distributi
on%20Department']"/>

  </event>

</subProcess>

</bp:ExchangeProcess>

```

BIBLIOGRAPHIE

- Adams, J., Koushik S, G. Vasudeva et G. Galambos. 2001. *Patterns for e-business: A Strategy for Reuse*. Mc Press.
- Albani, A., A. Keiblinger, K. Turowski et C. Winnewisser. 2003. « Domain Based Identification and Modelling of Business Component Applications ». In *7th East European conference on advances in databases and information systems*, Dresden, Allemagne. p. 30-45. Springer-Verlag.
- Andersen, J., E. Elsborg, F. Henglein, J. G. Simonsen et C. Stefansen. 2006. « Compositional specification of commercial contracts ». *International Journal on Software Tools for Technology Transfer*. Volume 8, no. 6, p. 485-516. Springer-Verlag.
- Andersson, B., M. Bergholtz, A. Edirisuriya, T. Ilayperuma, P. Johanesson, B. Gregoire, M. Schmitt, E. Dubois, S. Abels, A. Hahn, J. Gordijn, H. Weigand et B. Wangler. 2006. « Towards a Common Ontology for Business Models ». In *Proceedings of the 3rd Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability*. EMOI-INTEROP. vol. 200. En ligne. <<http://www.ceur-ws.org/Vol-200/10.pdf>>
- Andersson, B., M. Bergholtz, A. Edirisuriya, T. Ilayperuma, P. Johanesson et J. Zdravkovic. 2007. « On the alignment of goal models and business models ». In *REA-25. A Celebration of the REA Enterprise Model*. University of Delaware. En ligne. <<http://www.aisvillage.com/rea25/program.html>>
- Bae, J., L. Liu, J. Caverlee et W. B. Rouse. 2006. « Process mining, discovery, and integration using distance measures ». In *ICWS'06*, p. 479-488. IEEE Computer Society.
- Balogh, A., et D. Varro. 2006. « Advanced model transformation language constructs in the VIATRA2 framework ». In *Proceedings of the ACM symp. on Applied Computing*, Dijon. France, p. 1280-1287. ACM.
- Barbosa, P., F. Ramalho, J. Figueiredo, A. Junio, A. Costa et L. Gomes. 2009. « Checking Semantics Equivalence of MDA Transformations in Concurrent Systems ». *Journal of Universal Computer Science*, vol. 15, no. 11, p. 2196-2224.

- Batini, C., S. Ceri et S. B. Navathe. 1991. *Conceptual database design. An entity-relationship approach*. Addison Wesley.
- Bohrer, K. A. 1998. « Architecture of the San Francisco frameworks ». *IBM Systems Journal*, vol. 37, p. 156-169.
- Bruce, B. 1985. « Deep Case Systems for natural language understanding ». *Artificial Intelligence Technical Report*. no. 362. University of Illinois at Urbana-Champaign.
- Buhr, R. J. A. 1998. « Use Case Maps as Architectural Entities for Complex Systems ». *IEEE Transactions on Software Engineering*. vol. 24, no. 12, p. 1131-1155. IEEE Press.
- Carlson, W. M. 1979. « Business Information Analysis and Integration Technique - the new horizon ». *ACM SIGMIS Data Base*, vol.10, no.4, p. 3-9. ACM.
- Ciuksys, D., et A. Caplinskas. 2006. « Modelling of reusable business processes: an ontology-based approach ». *Advances in Information Systems Development*, vol. 1, p. 71-82. Springer-Verlag.
- . 2007. « Ontology-based approach to reuse of business process knowledge ». *Process Engineering Journal*, ISSN 1392-0561, Informacijos Mokslai 42-43, p. 168-174.
- Coad, P., D. North et M. Mayfield. 1997. *Object Models: Strategies, Patterns and Applications*. 2nd ed. Yourdon Press Computing Series, Prentice Hall.
- Coad, P., E. Lefebvre et J. De Luca. 1999. *Java Modeling in Color with UML: Enterprise Components and process*. Prentice Hall.
- Corradini, A., U. Montanari, F. Rossi, H. Ehrig, R. Heckel et M. Loewe. 1996. « Algebraic Approaches to Graph Transformation, Part I: Basic Concepts and Double Pushout Approach ». *Rapport technique*. TR-96-17. Université de Pise.
- Curran, T. A., G. Keller et A. Ladd. 1998. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Prentice Hall.
- Curtis, B., M. I. Kellner et J. Over. 1992. « Process modeling ». *Communications ACM*, 35, vol.35, no. 9, p. 75-90. ACM New York, NY, USA.
- Davenport, T. H., et J. Short. 1990. « The new industrial engineering: information technology and Business Process Redesign ». In *Sloan Management Review*, vol. 31, no. 4, p. 11-27. Routledge.
- Davenport, T. H. 1993. *Process innovation: reengineering work through information technology*. Harvard Business School Press.

- David, J. S. 1997. « Three “Events” That Define an REA Methodology for Systems Analysis, Design, and Implementation ». In *Proceedings of the Annual Meeting of the American Accounting Association*, Dallas, TX.
- Davis, A. M. 1993. *Software Requirements: Objects, Functions, and States*. Englewood Cliffs, Prentice Hall.
- Dunn, C., J. O. Cherrington et A. Hollander. 2004. *Enterprise Information Systems: A Pattern-Based Approach*. Irwin/mcgraw-hill.
- Eclipse. 2008. Eclipse Modeling Framework Project (EMF). Eclipse Foundation. <<http://www.eclipse.org/modeling/emf/>>. Consulté en avril 2012.
- El boussaidi, G. 2009. « Développement logiciel par transformation de modèles ». Thèse de Doctorat, Université de Montréal.
- Eriksson, H. E, et M. Penker. 2000. *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons.
- ESSI. 2006. Semantics Utilized for Process Management within and between Enterprises (SUPER). ESSI (European Semantic Systems Initiative). En ligne. <<http://www.ip-super.org>>. Consulté en mars 2012.
- Fenton, N. E., et S. L. Pfleeger. 1998. *Software metrics: a rigorous and practical approach*. PWS Publishing Co, 2nd Edition.
- Fillmore, C. J. 1968. « The Case for Case ». In *Bach & Harms (Ed.): Universals in Linguistic Theory*. New York: Holt, Rinehart, and Winston.
- . 1971. « Some problems for Case Grammar ». In: *R.J. O'Brien (Ed.): 22th Annual Round Table. Linguistics: Developments of the sixties – viewpoints of the seventies, Vol. 24 de Monograph Series on Language and Linguistics*, Georgetown University Press, Washington D.C.
- Fowler, M. 1997. *Analysis Patterns – Reusable Object Models*. Addison-Wesley.
- . 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd Edition, Addison-Wesley.
- Fox, M. S. 1992. « The TOVE Project: A Common-sense Model of The Enterprise ». In *Proceedings of the 5th international conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, p. 25-34. Springer-Verlag London, UK
- Fox, M. S., et M. Gruninger. 1998. « Enterprise Modeling ». *The American Association for Artificial Intelligence*, p. 109-121.

- Gailly, F., et G. Poels. 2007. « Ontology-Driven Business Modelling: Improving the Conceptual Representation of the REA Ontology ». In *Proceeding ER'07 Proceedings of the 26th international conference on Conceptual modeling*, p. 407-422. Springer-Verlag.
- Gamma, E., R. Helm, R. Johnson et J. Vlissides. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Geerts, G. L., et W. E. McCarthy. 2000. « The Ontological Foundations of REA Enterprise Information Systems ». In *American Accounting Association Conference*. Philadelphia.
- . 2001. « Using Object Templates from the REA Accounting Model to Engineer Business Processes and Tasks ». *The Review of Business Information Systems*, vol. 5, no. 4, p. 89-108.
- . 2002. « An ontological analysis of the economic primitives of the extended-REA enterprise information architecture ». *International Journal of Accounting Information Systems*, vol. 3, no. 1, p. 1-16.
- Geyer-Schulz, A., et M. Hahsler. 2001. « Software engineering with analysis patterns ». *Technical Report*. Wirtschaftsuniversität Wien.
- Gordijn, J., H. Akkermans et H. Van Vliet. 1999. « Requirements for e-commerce applications are created rather than elicited ». In *Proceedings of the second nordic workshop on software architecture- NOSA'99*.
- . 2000. « What's in an Electronic Business Model? ». In *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th International Conference, EKAW 2000*, vol. 1937, p. 257-273. Springer Berlin Heidelberg.
- Gordijn, J., et H. Akkermans. 2001. « Designing and evaluating EBusiness models ». *IEEE Intelligent Systems*. vol. 16, no. 4, p. 11-17.
- Gruber, T. R. 1993. « Towards Principles for the Design of Ontologies Used for Knowledge Sharing ». *International Journal of Human-Computer Studies*. vol. 43. no. 5-6, P. 907-928.
- Gunther, C. W., S. Rinderle, M. Reichert et W. M. P. van der Aalst. 2006. « Change mining in adaptive process management systems ». In *CoopIS'06*, pages 309-326. Springer-Verlag Berlin, Heidelberg.
- Hallerbach, A., T. Bauer et M. Reichert. 2008. « Managing process variants in the process life cycle ». *Information Systems Journal*, vol. 2, p. 154-161.

- . 2010. « Capturing variability in business process models: the Provop approach ». *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 22, no. (6-7), p. 519-546. Wiley InterScience.
- Hammer, M., et J. Champy. 1993. *Reengineering the Corporation : A Manifesto For Business Revolution*. Harper Business. New York.
- Heckel, R., J. M. Küster et G. Taentzer. 2002. « Confluence of Typed Attributed Graph Transformation Systems ». In *Proceedings of the First International Conference on Graph Transformation*, p. 161-176. Springer-Verlag.
- Hepp, M., et D. Roman. 2007. « An Ontology Framework for Semantic Business Process Management ». In *Proceedings of Wirtschaftsinformatik*, Karlsruhe.
- Hruby, P. 2006. *Model-Driven Design Using Business Patterns*. Springer-Verlag.
- IBM. 2009. Dynamic Business Process Management (DBPM). IBM Corporation.
- Ijiri, Y. 1975. *Theory of Accounting Measurement*. American Accounting Association.
- Ilayperuma, T. 2007. « Reference Ontology for Business Models ». PhD Thesis. Computer and Systems Sciences Stockholm University, Sweden.
- ISO/IEC. 2010. Business operational aspects of open-edi for implementation. ISO/IEC FDIS 15944-1.
- JBoss. 2012. JBoss Drools (Drools). En ligne. <<http://www.jboss.org/drools>>. Consulté en mars 2012.
- Jaquet, M. 2006. « A Property Driven Approach towards Describing Semantics of REA Entities ». In *Proceedings of the 2nd International REA Technology Workshop*. Santorini Island, Greece, p. 31-43. IT University of Copenhagen.
- Johannesson, P. 1993. « Using Conceptual Graph Theory to Support Schema Integration ». In *12th International Conference on Entity-Relationship Approach*, vol. 823, p. 283-296. Springer Berlin Heidelberg.
- Johansson, H. J., P. McHugh, A. J. Pendlebury et W. A. Wheeler. 1993. *Business Process Reengineering: BreakPoint Strategies for Market Dominance*. John Wiley & Sons.
- Ko, R. K. L. 2009. « A computer scientist's introductory guide to business process management », *Crossroads*, vol. 15, no. 4, p. 11-18. ACM.
- Ko, R. K. L., S. S. G. Lee et E. W. Lee. 2009. « Business Process Management Standards: A Survey ». In *Business Process Management Journal*, vol. 15, no. 5, p. 744-791. Emerald Group Publishing Limited.

- Küster, J. M. 2006. « Definition and validation of model transformation ». *Software and Systems Modeling (SoSyM)*, vol 5, no. 3, p. 233-259. Springer-Verlag.
- Küster, J. M., et M. Abd-El-Razik. 2006. « Validation of Model Transformations – First Experiences using a White Box Approach ». In *Proceedings of the 2006 international conference on Models in software engineering*, p. 193-204. Springer-Verlag
- Lano, K., et J. Bicarregui. 1999. « Semantics and Transformations in UML Models ». In *UML 98: Beyond the Notation, Lecture Notes in Computer Science*, vol. 1618, p. 107-119. Springer-Verlag.
- Li, C., M. Reichert et A. Wombacher. 2008. « Discovering reference process models by mining process variants ». In *Proceedings of the 2008 IEEE International Conference on Web Services*, p. 45-53. IEEE Computer Society Press.
- Lubars, M., G. Meredith, C. Potts et C. Richter. 1992. « Object-Oriented Analysis for Evolving Systems ». In *Proceedings of the 14th International Conference on Software Engineering*, p. 173-185. ACM Press.
- Magal, S. R., et J. Word. 2010. *Integrated Business Processes With ERP Systems*. Wiley; 2nd Edition.
- Malone, T. W., K. Crowston, J. Lee, B. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C.S. Osborn, A. Bernstein, G. Herman, M. Klein et E. O'Donnell. 1999. « Tools for inventing organizations: Toward a handbook of organizational processes ». *Management Science*, vol. 45, no. 3, p. 425-443.
- Malone, T. W., K. Crowston et G. A. Herman. 2003. *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press.
- Marschall, F., et P. Braun. 2003. « Model Transformations for the MDA with BOTL ». In *Proceedings of the workshop on Model Driven Architecture : Foundations and Application*, p. 25-36. University of Twente.
- McCarthy, W. E. 1982. « The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment ». *The Accounting Review* (July 1982), vol. 57, no. 3, p. 554-578.
- . 2004. « An REA Model of an Economic Exchange ». *Research Seminar on REA Modeling and Design Science*. Michigan State University.
- Miftah, H. 2010. « Business Blueprint Best Practice – SAP MATERIAL MANAGEMENT ». *Rapport technique*.
- Mili, H., F. Pachet. 2000. « Metamodeling for Multidimensional Reuse ». *Proceedings of the Maghrebien Conference on Software Engineering and AI (MCSEAI'2000)*, p. 29-39.

- Mili, H., A. Mili, S. Yacoub et E. Addy. 2001. *Reuse-Based Software Engineering : Techniques, Organizations, and Measurement*. Wiley-Interscience.
- Mili, H., G. Bou-Jaoude, E. Lefebvre et G. Tremblay. 2004. « Going beyond MDA: Business Process Modeling for Software Reuse ». *OOPSLA'2004*, Canada, p. 1-15.
- Mili, H., P. Valtchev, A. Leshob, A. Obaid et G. Lévesque. 2007. « Towards Building software systems from a specification of the business processes that they support ». In *Proceedings of the 7th OOPSLA Workshop on Domain-Specific Modeling*, Montreal, Canada.
- Mili, H., A. Leshob, E. Lefebvre, G. Lévesque et G. El Boussaidi. 2009. « Towards a methodology for representing and classifying business processes ». In *Proceedings of the 4th International MCETECH Conference on e-Technologies*, Ottawa, Canada, p. 196-211. Springer Berlin Heidelberg.
- Mili, H., G. Tremblay, G. Bou-Jaoude, E. Lefebvre et L. Elabed. 2010. « Business Process Modeling Languages: Sorting Through the Alphabet Soup ». *ACM Computing Surveys*, vol. 43, no. 1, Article 4. ACM.
- Mili, H., P. Valtchev, Y. Charif, L. Szathmary, N. Daghrir, M. Béland, L. Martin, F. Bédard, S. Caid-Essebsi et A. Leshob. 2011. « E-Tourism Portal: a case study in ontology-driven development ». In *5th International Conference, MCETECH*, p. 76-99. Springer Berlin Heidelberg.
- Murray, M. 2009. *Discover Logistics with SAP ERP*. SAP PRESS, 1st edition.
- OMG. 2003a. Model Driven Architecture (MDA). OMG .
- . 2003b. XML Metadata Interchange Specification (XMI). OMG.
- . 2008. Business Process Definition Metamodel (BPDM). OMG.
- Osterwalder, A., et Y. Pigneur. 2002. « An ebusiness model ontology for modeling eBusiness ». In *Proceedings of 15th Bled Electronic Commerce Conference*, Paper 2. Slovenia.
- Osterwaldor, A. 2004. « The business model ontology ». Thèse de Doctorat, École des Hautes Études Commerciales, Université de Lausanne.
- Ould, M. A. 1995. *Business Processes: Modelling and Analysis for Re-engineering and Improvement*. John Wiley & Sons.
- Pateli, A. G., et G. M. Giaglis. 2004. « A research framework for analysing eBusiness models ». *European Journal of Information Systems*, vol.13, no. 4, p. 302-314. Macmillan Press Ltd.

- Phalp, K., et M. Shepperd. 2000. « Quantitative analysis of static models of processes ». *The Journal of Systems and Software*. vol. 52, no. 2-3, p. 105-112. Elsevier Inc.
- Polya, G. 1982. *How to Solve It: A New Aspect of Mathematical Method*. Princeton Science Library.
- Porter, M. E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press.
- Pressman, R. S. 2005. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Higher Education.
- Recker, J. 2005. « Process Modeling in the 21st Century ». In *BP Trends*. vol. 3. p. 1-6.
- Rosemann, M., et W. M. P. van der Aalst. 2003. « A Configurable Reference Modelling Language ». *Technical report*, Queensland University of Technology, Brisbane.
- RosettaNet. 2010. « eBusiness Standards for the Global Supply Chain ». RosettaNet. En ligne. <<http://www.rosettanet.org/>>
- . 2011. « Overview, clusters, segments, and pips ». RosettaNet. En ligne. <<http://www.rosettanet.org/TheStandards/RosettaNetStandards/PIPOverview/tabid/3482/Default.aspx/>>
- Rubin, B. S., A. R. Christ et K. A. Bohrer. 1998. « Java and the IBM San Francisco project ». *IBM Systems Journal*, vol. 37, p. 365-371.
- Schuster, R., et T. Motal. 2009. « From e3-value to REA: Modeling multi-party eBusiness Collaborations ». In *IEEE Conference on Commerce and Enterprise Computing*. Vienna, Austria, p.202-208. IEEE Computer Society.
- Shlaer, S., et S. J. Mellor. 1992. *Object Lifecycles: Modeling the World in States*. Yourdon Press.
- UN/CEFACT. 2005. Catalog of Common Business Processes. *UN/CEFACT Technical Specification*. Version 1.0
- Ushold, M., et M. Gruninger. 1996. « Ontologies: Principles, Methods and Applications ». *Journal of The Knowledge Engineering Review*, vol. 11, no. 2, p. 93-136.
- Ushold, M., M. King, S. Moralee et Y. Zorgios. 1998. « The Enterprise Ontology ». In *The Knowledge Engineering Review*, vol. 13.
- Varro, D., G. Varro et A. Pataricza. 2002. « Designing the automatic transformation of visual languages ». *Special issue on applications of graph transformations (GRATRA)*, vol. 44, n.2, p. 205-227. Elsevier North-Holland, Inc.

- Varro, D., et A. Pataricza. 2003. « Automated Formal Verification of Model Transformations ». In *Critical Systems Development in UML 2003*, p. 63-78. Technische Universität München.
- WfMC. 1999. Interface 1: Process Definition Interchange Process Model. WfMC-TC-1016-P, Version 1.1.
- Wiegers, K. E. 2003. *Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle*. Microsoft Press.
- Wohed, P. 2000. « Conceptual Patterns for Reuse in Information Systems analysis ». *Advanced Information Systems Engineering: 12th International Conference, CAiSE 2000*, Stockholm, Sweden, vol.1789, p. 157-175. Springer-Verlag.
- . 2001. « Conceptual Patterns- A Consolidation of Coad's and Wohed's Approaches ». In *Proc. of the 5th International Conference on Applications of Natural Language to Information Systems*, vol.1959, p. 340-351. Springer-Verlag.
- Wyner, G. M., et L. Jintae. 2003. « Defining Specialization for Process Models ». In *Malone et al. (2003)*, P. 131-174. MIT Press.